②

NAVAL POSTGRADUATE SCHOOL
Monterey , California

AD-A226 931

DTIC
ELECTE
SEP. 27. 1990
S B D

THESIS

NUMERICAL STUDIES OF COMPRESSIBLE FLOW
OVER A DOUBLE-DELTA WING AT
HIGH ANGLES OF ATTACK

by

Raymond L. Coutley

March 1990

Thesis Advisor: M. F. Platzer
Co-Advisor: J. A. Ekaterinaris

---

90 09 27 044

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report
2b Declassification Downgrading Schedule	Approved for public release; distribution is unlimited.

4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)

6a Name of Performing Organization Naval Postgraduate School	6b Office Symbol *(if applicable)* 31	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000		7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000

8a Name of Funding Sponsoring Organization	8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*		10 Source of Funding Numbers

Program Element No	Project No	Task No	Work Unit Accession No

11 Title *(Include security classification)* NUMERICAL STUDIES OF COMPRESSIBLE FLOW OVER A DOUBLE-DELTA WING AT HIGH ANGLE OF ATTACK (Unclassified)

12 Personal Author(s) Raymond L. Coutley

13a Type of Report Master's Thesis	13b Time Covered From To	14 Date of Report *(year, month, day)* March 1990	15 Page Count 171

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	word processing, Script, GML, text processing.

19 Abstract *(continue on reverse if necessary and identify by block number)*

The objective of this work is the investigation of vortical flows at high angles of attack using numerical techniques. First step for a successful application of a numerical technique, such as finite difference or finite volume, is the generation of a computational mesh which can capture adequately and accurately the important physics of the flow. Therefore, the first part of this work deals with the grid generation over a double-delta wing and the second part deals with the visualization of the computed flow field over the double-delta wing at different angles of attack. The surface geometry of the double-delta wing is defined algebraically. The developed surface grid generator provides flexibility in distributing the surface points along the axial and circumferential directions. The hyperbolic grid generation method is chosen for the field grid generation and both cylindrical and spherical grids are constructed. The computed low speed ($M = 0.2$) flow results at different angles of attack over the double-delta wing are visualized. Important flow characteristics of the leeward side flow field are discussed while the development of vortex interaction, occurrence and progression of vortex breakdown as the angle of attack increases is demonstrated. The computed results at different fixed angles of attack are presented.

20 Distribution Availability of Abstract ☒ unclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual M. F. Platzer	22b Telephone *(include Area code)* (408) 646-2058	22c Office Symbol 67Pl

Numerical Studies of Compressible Flow Over
a Double-Delta Wing at High Angle of Attack

by

Raymond L. Coutley
Lieutenant, United States Navy
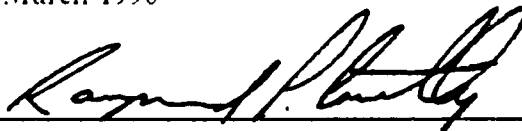B.S., Marquette University, 1978

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING
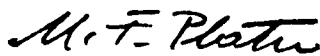and
AERONAUTICAL ENGINEER
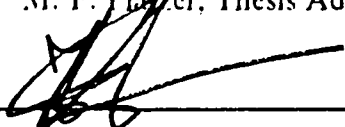
from the

NAVAL POSTGRADUATE SCHOOL
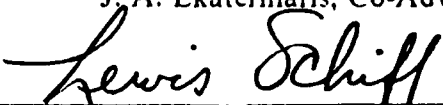March 1990

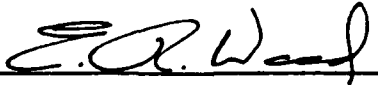Author: _____

Raymond L. Coutley

Approved by: _____

M. F. Platzer, Thesis Advisor

_____

J. A. Ekaterinaris, Co-Advisor

_____

L. B. Schiff, Second Reader

_____

E. R. Wood, Chairman,
Department of Aeronautics and Astronautics

_____

Dean of Faculty and Graduate Studies

ii

ABSTRACT

The objective of this work is the investigation of vortical flows at high angles of attack using numerical techniques. First step for a successful application of a numerical technique, such as finite difference or finite volume, is the generation of a computational mesh which can capture adequately and accurately the important physics of the flow. Therefore, the first part of this work deals with the grid generation over a double-delta wing and the second part deals with the visualization of the computed flow field over the double-delta wing at different angles of attack. The surface geometry of the double-delta wing is defined algebraically. The developed surface grid generator provides flexibility in distributing the surface points along the axial and circumferential directions. The hyperbolic grid generation method is chosen for the field grid generation and both cylindrical and spherical grids are constructed. The computed low speed ($M = 0.2$) flow results at different angles of attack over the double-delta wing are visualized. Important flow characteristics of the leeward side flow field are discussed while the development of vortex interaction, occurrence and progression of vortex breakdown as the angle of attack increases is demonstrated. The computed results at different fixed angles of attack are presented.

iii

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I. INTRODUCTION

The objective of this work is the investigation of vortical flows over three-dimensional bodies at high incidence utilizing numerical methods. The advantage of numerical simulations compared with experiment is that they allow simultaneous observation of all flow quantities of interest for the entire flow field. The disadvantage of numerical techniques is the accuracy limitations for simulations of flow fields over complex realistic configurations, even with the most efficient numerical schemes and fast computers. High Reynolds number turbulent flows of engineering interest can be fully simulated when all relevant scales are resolved. Resolution of all scales for complex high reynolds number flows over realistic configurations is beyond the capabilities of the present and next generation supercomputers. Common practice for the simulation of engineering flows is the use of various turbulence models to approximate the effect of the small scales which cannot be resolved. Error sources in numerical simulations are related to the discretization process, the order of accuracy of the numerical scheme and the turbulence modeling that is used.

Nevertheless. Computational Fluid Dynamics (CFD) allows investigation of various fluid flow phenomena that in the past was possible only in wind tunnels, water tunnels or actual flight testing. The advantage of being able to accurately capture the flow characteristics over complex configurations or even complete aircraft without endangering life, i.e. preliminary flight testing, is readily apparent. Numerical solutions also enable to investigate and visualize the flow field characteristics from any viewpoint or in as much detail as desired. With the ever increasing speed cost ratio of today's computers. CFD techniques will be playing a more significant role facilitating aerodynamic research and supplementing experimental investigations. Even though CFD and Navier-Stokes methods are not a new research tool, new and more efficient numerical techniques are evolving, while at the same time computers are becoming faster. Numerical prediction of steady flows over complete aircraft and comparison with flight data is already underway [Ref. 1]. In the near future CFD is expected to play a more active role in fluid dynamic research enabling simulation of complex unsteady flow regimes.

In the past panel methods and vortex lattice methods were used in the analysis of flows [Ref. 2]. These methods were insufficient for a detailed analysis of complex flows such as vortical flows over bodies at incidence. The limitations of these methods are due

1

to the potential flow assumption which is valid only for inviscid and irrotational flow. Viscous effects close to the surface for attached or mildly separated flow are obtained using Boundary Layer methods [Ref. 3]. The rotational compressible flow regime at high Reynolds numbers was investigated with the Euler equations. Viscous effects become more important for flows at high angles of attack; therefore, the solution of the Navier-Stokes equations is required for this flow regime.

In Chapter 2 the theoretical development of the compressible Navier-Stokes equations will be discussed. A finite difference algorithm used for the numerical solution of these equations will be presented. The numerical solution is performed on the finite number of points obtained after discretization of the flow domain. The procedure which yields this finite collection of points in the solution domain is known as grid generation. The quality of the solution depends directly on the smoothness of the grid and its ability to accurately represent flow gradients. Therefore, grid generation is an important part of the numerical solution. However, the numerical solution of the governing equations is not the main objective of this research. The grid generation part, which is a necessary stage before starting the numerical solution will be covered in full detail. Numerical solutions depend on the representation of the flow field by an orderly, finite collection of points. The process of obtaining three-dimensional grids involves first definition of an inner boundary, commonly known as the surface grid, before the subsequent generation of the field grid can begin.

The methods available for both the surface and field grid generation will be covered in Chapters 3 and 4, respectively. Developments in the area of grid generation have provided a key to eliminate the problem of boundary shape definition [Ref. 4]. Finite difference grids can also be used to construct meshes that are suitable in finite element methods. The specific numerical method utilized in this research is the finite difference method. The finite difference method is one of the oldest numerical methods that can be utilized to obtain numerical solutions to differential equations. The application of this method is based on a Taylor series expansion and the definition of the derivative; most likely first developed by Euler in 1768 [Ref. 5: p. 167]. The algorithm used for the numerical integration utilizes a partially flux-split numerical scheme with central differencing in the other two directions [Ref. 6].

The methods described above will be applied to a double-delta wing that has a strake with a sweep angle of 76° and a delta wing section with a sweep angle of 40° . Particular emphasis will be placed on the investigation of the vortical flow field at moderate to high angles of attack. Separated flow along the strake's leading edge forms free shear layers

2

which roll up to form vortex cores. This primary strake vortex generates an additional non-linear lift called vortex induced lift. A primary wing vortex also develops from the leading edge of the 40° swept delta wing.

The mutual interaction of the strake and wing vortices and their interaction with the surface is an active area of current research. Investigation and prediction of the vortex breakdown that appears at higher angles of attack is also of high interest. The development of the leading edge vortex as well as breakdown are important phenomena that need to be fully understood. Various angles of attack, $\alpha = 10.0°$, $19.0°$, $22.4°$ are investigated and compared with available experimental data. Understanding the leeward-side flow structure as well as breakdown are important phenomena that affect significantly today's tactical and fighter aircraft effectiveness.

Vortex breakdown is a transition of the vortex core from a jet-like flow to a wake-like flow. Both swirl angle and adverse pressure gradient along the axial direction contribute to the breakdown of the vortex. Peckham and Atkinson first identified vortex breakdown when analyzing delta wings at high angles of attack [Ref. 7]. Research on vortex breakdown was continued by Elle, Lambourne and Bryer, Harvey, Pritchard, Sarpkaya, Hummel, Faler and Leibovitch, Payne and Nelson [Ref. 8,9,10,11, 12 ,13,14,15,16]. Studies then naturally progressed to more complicated bodies such as the double-delta wing where Brennenstuhl tested several wings in a low speed wind tunnel and a water tunnel [Ref. 17]. The present study will attempt a comparison of the computational solution with the data obtained from wind tunnel testing done by Cunningham and Boer [Ref. 18]. This comparison along with discussions of the results that were developed during this research will be covered in Chapter 5. The closing chapter summarizes the conclusions and presents recommendations for further research.

3

II. THEORETICAL APPROACH

The main objective of this work is the investigation of different techniques for the grid generation over complex three-dimensional bodies, and numerical flow visualization of the computed flowfields over bodies at high angles of attack. The flow field is obtained by the numerical solution of the Navier-Stokes equations. Fluid flow in the continuum flow regime includes most of the physical flows and is governed by the Navier-Stokes equations. The derivation of the Navier-Stokes equations is well known [Ref. 3: pp. 47-66]. Solutions of these equations are of interest in basic fluid mechanics research and for engineering applications. The solution of the Navier-Stokes equations is quite difficult due to their nonlinearity. Analytical closed form solutions of the Navier-Stokes equations can be obtained for only a few flow situations of simple geometrical configurations and boundary conditions. Simplified forms of the Navier-Stokes equations, such as the boundary layer equations, can give satisfactory answers for many flows of practical interest. The inviscid form of the Navier-Stokes equations, commonly known as the Euler equations, can provide solutions for flows away from solid boundaries. However, complex flows such as vortical separated flows require the solution of the full Navier-Stokes equations, which can only be obtained by utilizing numerical techniques. The derivation of the Navier-Stokes equations is outlined in the following paragraphs.

A. GOVERNING EQUATIONS

1. The Continuity Equation

For the derivation of the Navier-Stokes equations the fluid medium is considered as an isotropic, homogeneous, compressible and viscous Newtonian fluid. The continuity equation is a manifestation of the fact that mass can neither be created nor destroyed. The continuity equation states that the time variation of density within a control volume plus the mass entering and leaving the control volume is equal to zero. The differential form of the continuity equation for a compressible fluid and non-steady flow can be written as;

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V}) = 0. \tag{1}$$

For low speeds the density variation is small, therefore;

$$\frac{\partial \rho}{\partial t} = 0$$

and,

$$\vec{\nabla} \cdot (\rho \vec{V}) = \rho \vec{\nabla} \cdot \vec{V}.$$

Hence, for incompressible flow, the continuity equation can be written as;

$$\vec{\nabla} \cdot \vec{V} = 0.$$

2. Derivation of the Navier-Stokes Equations

For a compressible fluid, all primitive variables, density (ρ), pressure (p) and velocity ($\vec{V}$) are functions of space and time. Newton's Second Law states that the summation of all forces must be equal to the mass times the acceleration.

$$\sum \vec{F} = M\vec{a} \tag{2}$$

Considering an infinitesimally small fluid particle or control volume moving in a Cartesian Coordinate System, the right-hand side of equation (1) can be rewritten as;

$$M\vec{a} = \frac{\mathbf{D}}{\mathbf{D}t} (\rho \vec{V}) dx dy dz \tag{3}$$

where $\dfrac{\mathbf{D}}{\mathbf{D}t}$ is the material derivative.

$$\frac{\mathbf{D}}{\mathbf{D}t} (\rho \vec{V}) = \rho \frac{\mathbf{D}}{\mathbf{D}t} \vec{V} + u \frac{\mathbf{D}}{\mathbf{D}t} \rho$$

and $\vec{V}$ is the velocity vector, which for a Cartesian Coordinate system is,

$$\vec{V} = u\vec{i} + v\vec{j} + w\vec{k} \tag{4}$$

here u, v, w are the velocity components along the coordinate axes. The external forces normally consist of the gravitational forces and the forces acting on the boundaries of the control volume, namely pressure and friction. All other body forces, such as electromagnetic forces will be ignored. For simplicity, the momentum equation only for the x-direction will be derived, while the derivation is analogous for the y and z-directions. The x-component of equation (2) is;

$$M\vec{a}_x = \frac{\mathbf{D}}{\mathbf{D}t}(\rho u)dxdydz. \tag{5}$$

Figure (1) shows the normal and shear stresses on an infinitesimal control volume. Summation of the forces in the x-direction yields;

$$Surface\ Forces = \sigma_x dydz + (\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y}dy)dxdz + (\tau_{zx} + \frac{\partial \tau_{zx}}{\partial z}dz)dxdy$$

$$- (\sigma_x + \frac{\partial \sigma_x}{\partial x}dx)dydz - \tau_{zx}dxdy - \tau_{yx}dxdz$$

which reduces to,

$$Surface\ Forces = (-\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z})dxdydz. \tag{6}$$

Among the body forces only the gravity will be considered. Therefore, if f_x is the x-component of the gravity force then;

$$Weight = f_x(x,y,z)\rho(x,y,z)dxdydz. \tag{7}$$

The sum of equations (6) and (7) are the external forces which are equal to the acceleration as stated by equation (5). After cancellation of the common term of volume $(dx\ dy\ dz)$, the following force balance for the x-direction is obtained.

$$\frac{\mathbf{D}}{\mathbf{D}t}(\rho u) = \rho f_x + (-\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}) \tag{8}$$

The next step is to express the stresses in terms of the primitive variables, i.e., velocities and pressure. First, the static pressure is defined as the mean of the normal stresses.

$$p = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z)$$

This equation can be algebraically rewritten as;

$$\sigma_x = p + \frac{1}{3}(2\sigma_x - \sigma_y - \sigma_z). \tag{9}$$

6

Figure 1. Normal and Shear Stress due to Friction

In equation (9) the left-hand side is the normal stress at a point in the fluid. The first right-hand side term is the static pressure and the second right-hand side term is the deviation of the normal stress from the pressure due to viscous forces. Next a relationship between stress and rate of strain must be found. Isotropy implies that this relation between the components of stress and rate of strain is the same for every direction. The Newtonian fluid assumption means that this relationship is also linear. Referring to Figure (2) where σ_x and σ_y are resolved into diagonal components and equating the forces, the following force balance is obtained.

$$\tau'_{xy}(a \sqrt{2}) + \sigma_x(\frac{a}{\sqrt{2}}) - \sigma_y(\frac{a}{\sqrt{2}}) = 0$$

This equation can be rewritten as:

$$-\tau'_{xy} = \frac{1}{2}(\sigma_x - \sigma_y). \qquad (10)$$

7

Figure 2.   Normal and Shear Stress

A similar equation can be derived for the xz-plane.

$$-\tau'_{zx} = \frac{1}{2}(\sigma_z - \sigma_x) \tag{11}$$

Substitution of equations (10) and (11) into equation (9) results in;

$$\sigma_x = p + \frac{2}{3}(\tau'_{zx} - \tau'_{xy}). \tag{12}$$

The deformation of the initial shape of the fluid element (ABCD) to (A'BC'D) due to stresses in the y-direction is shown in Figure (3). The same figure also shows that the length of OA' is as follows;

$$(O.A') = Length = \left(u + \frac{\partial u}{\partial x} \frac{a}{\sqrt{2}} + H.O.T.\right)\Delta t.$$

The length change due to stress is:

8
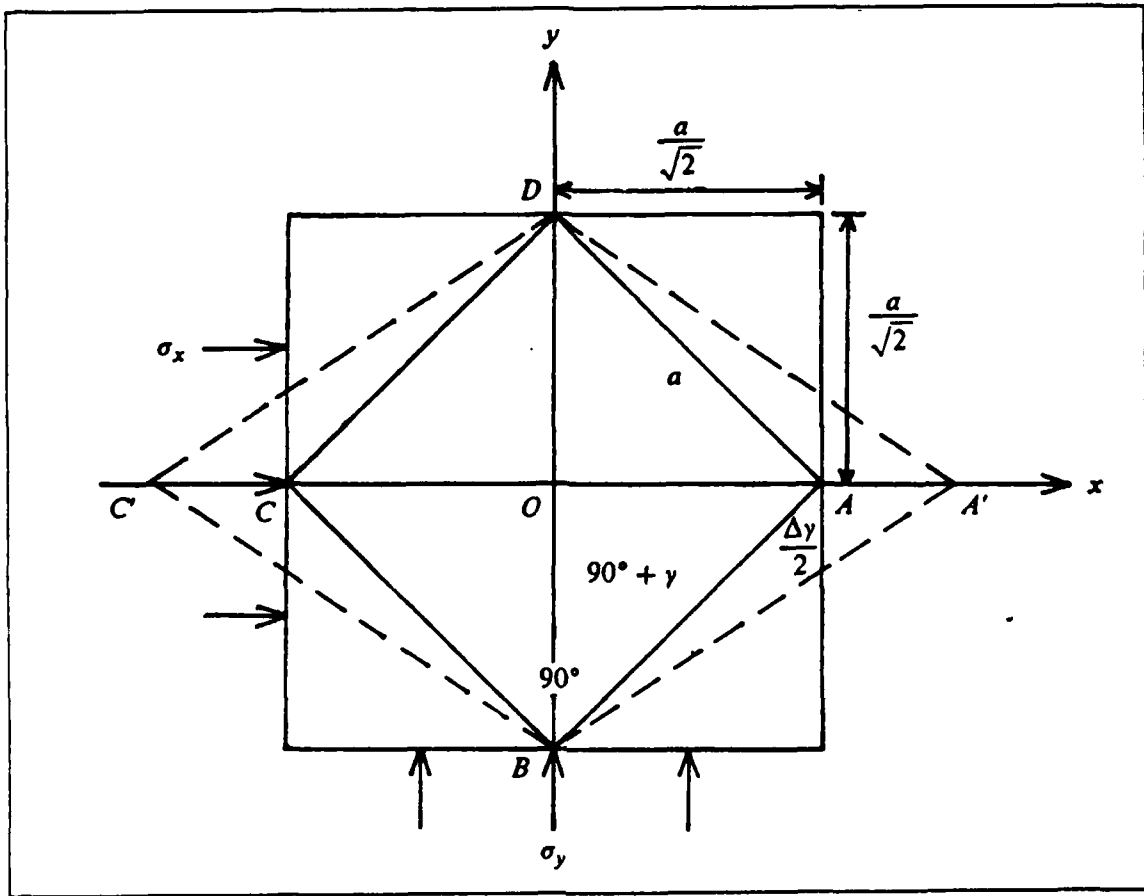
Figure 3. Rate of Strain

$$(AA') = \Delta Length \approx \frac{\partial u}{\partial x}\frac{a}{\sqrt{2}}\Delta t.$$

The strain is the change in length divided by the original length which is $a/\sqrt{2}$. The strain rate is obtained by dividing this length by Δt . The same procedure can be repeated for the y-direction to obtain the resultant rate of strain on a 45 degree plane due to σ_y . The shear stress on the 45° plane due to σ_x and σ_y is given by equation (13). A similar procedure for the zx-plane yields the analogous shear stress which is shown in equation (14).

$$\tau'_{xy} = \mu(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}) \qquad (13)$$

9

$$\tau'_{zx} = \mu \left(\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} \right) \tag{14}$$

In the above equations, the proportionality constant μ, is defined as the coefficient of viscosity. Substituting equation (13) and (14) back into equation (12), and using Stokes' Hypothesis;

$$3\lambda + 2\mu = 0$$

yields equation (15) [Ref. 3: pp. 60-61].

$$\sigma_x = p - \mu \left(2\frac{\partial u}{\partial x} - \frac{2}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right) \tag{15}$$

In this equation the first term in parenthesis is the linear strain rate and the second term is the volumetric strain rate. To complete the derivation. the terms τ_{xy} and τ_{zz} in equation (8) will be expressed in terms of the velocity components. From Figure (4) the rate of strain can be obtained [Ref. 19: p. 93].

The rate of strain on this element is $\frac{\Delta \gamma}{\Delta t}$. Assuming that the variation of the rate of strain (γ) is small, the following expressions can be written.

$$\Delta \gamma = \frac{\Delta \gamma}{2} + \frac{\Delta \gamma}{2}$$

$$\Delta \gamma \approx \frac{\frac{\partial u}{\partial y} \Delta y \Delta t}{\Delta y} + \frac{\frac{\partial v}{\partial x} \Delta x \Delta t}{\Delta x}$$

$$\Delta \gamma = \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \Delta t$$

Taking the limit of $\frac{\Delta \gamma}{\Delta t}$ as $\Delta t \rightarrow 0$, the rate of strain is given by:

$$\frac{d\gamma}{dt} = \tau_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} . \tag{16}$$

Due to isotropy. τ_{xy} is equal to τ_{yx}. Analogous procedures used to derive equation (16) can be repeated for the yz-plane and zx-plane, respectively, so that for a Newtonian Fluid equations (17) and (18) can be written.

10
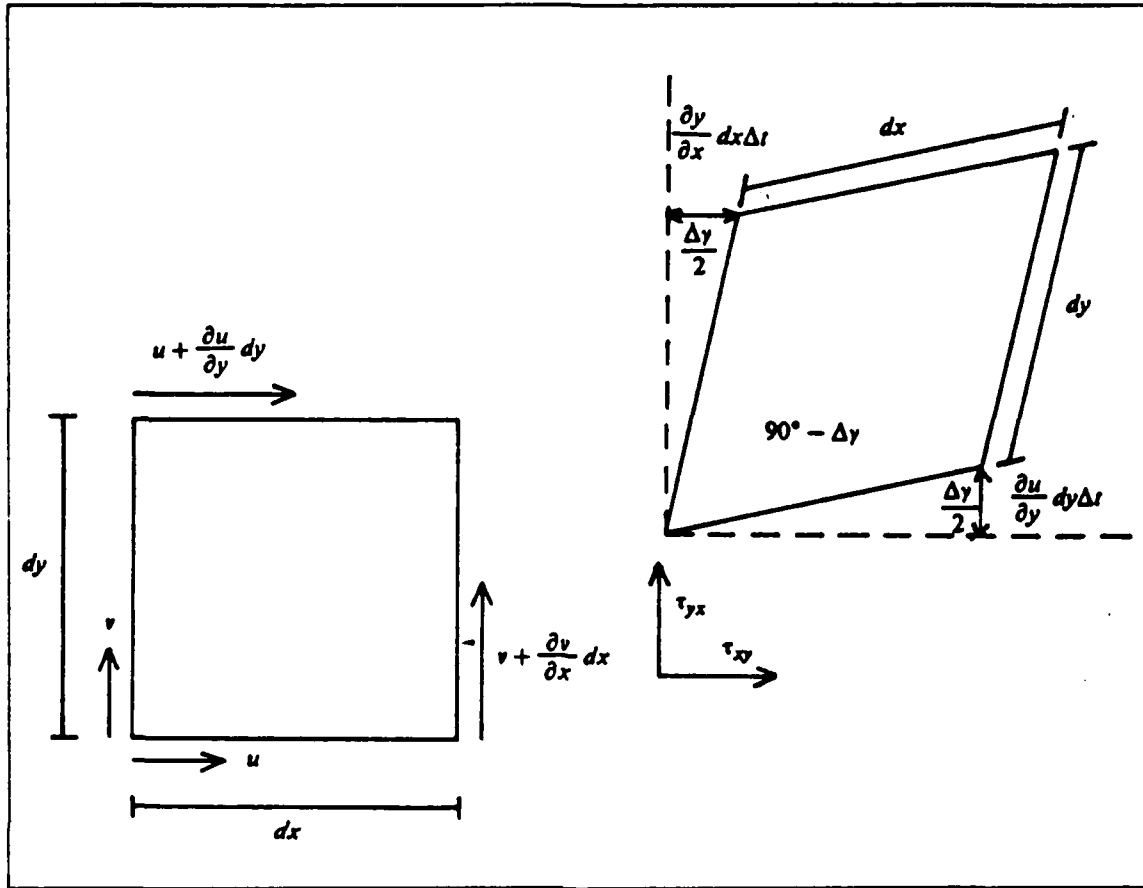
Figure 4.   Rate of Strain

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \qquad (17)$$

$$\tau_{zx} = \mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \qquad (18)$$

Finally, substituting equations (15), (17) and (18) back into equation (8), the momentum equation for the x-direction is obtained.

$$\frac{D}{Dt}(\rho u) = \rho f_x - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[\mu(2\frac{\partial u}{\partial x} - \frac{2}{3}\vec{\nabla}\cdot\vec{V}) \right]$$

$$+ \frac{\partial}{\partial y}\left[\mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \right] + \frac{\partial}{\partial z}\left[\mu(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}) \right] \qquad (19)$$

11

Similarly, the momentum equations for the y-direction and z-direction can be derived. These equations are;

$$\frac{D}{Dt}(\rho v) = \rho f_y - \frac{\partial p}{\partial y} + \frac{\partial}{\partial y}\left[\mu(2\frac{\partial v}{\partial y} - \frac{2}{3}\vec{\nabla}\cdot\vec{V})\right]$$

$$+ \frac{\partial}{\partial x}\left[\mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y})\right] + \frac{\partial}{\partial z}\left[\mu(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z})\right] \tag{20}$$

and,

$$\frac{D}{Dt}(\rho w) = \rho f_z - \frac{\partial p}{\partial z} + \frac{\partial}{\partial z}\left[\mu(2\frac{\partial w}{\partial z} - \frac{2}{3}\vec{\nabla}\cdot\vec{V})\right]$$

$$+ \frac{\partial}{\partial x}\left[\mu(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x})\right] + \frac{\partial}{\partial y}\left[\mu(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z})\right]. \tag{21}$$

The unknowns in these last three equations are the primary variables; the density, the velocities and the pressure. (ρ, u, v, w, p) . The momentum equations along with the continuity equation constitute the Navier-Stokes equations in the primitive variable formulation. The continuity equation for a Cartesian coordinate system is restated;

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0. \tag{22}$$

Here the pressure is related to the density through the equation of state.

$$p - \rho RT = 0 \tag{23}$$

For an isothermal process and incompressible flow, equation (19) through equation (23) would be sufficient, but when temperature variations depend on density and pressure, the energy equation is also required. This is always the case for compressible flow where density depends on pressure and temperature. The energy equation expresses the balance between heat and mechanical energy. The variation of viscosity due to temperature variation may be obtained by an empirical viscosity law. The final result is a system of five partial differential equations with five unknowns; u, v, w, p , and ρ .[Ref. 3, 19]

3. Derivation of the Energy Equation

It is well known that energy can be neither created nor destroyed but it can only change in form. Therefore an energy balance exists for a fluid element in motion. This

energy balance is obtained through certain mechanisms which for a compressible fluid are determined by changes in heat content, total energy and mechanical work. Changes in heat content can be due to convection, conduction, friction and/or radiation. For the following derivation radiation is neglected because its effect is small at moderate temperatures. The energy balance for a control volume is expressed by the first law of thermodynamics.

$$\frac{dQ}{dt} = \frac{dE}{dt} + \frac{dW}{dt} \tag{24}$$

First the variation of mechanical work or the contribution to work done by the external forces acting along the x-direction is derived. Again referring to Figure (1), the contribution to work done by each of the stress components is;

$$dW_{\sigma x} = -\left[-u\sigma_x + (u + \frac{\partial u}{\partial x} dx)(\sigma_x + \frac{\partial \sigma_x}{\partial x} dx) \right] dydz$$

which reduces to,

$$dW_{\sigma x} = -\left[\frac{\partial}{\partial x} (u\sigma_x) \right] dxdydz. \tag{25}$$

Continuing with the same procedure for the other components of shear stresses; the y-direction and the z-direction, the total change in work due to normal and shear stresses can be written as shown in equation (26).

$$dW =$$

$$-dV\left[\frac{\partial}{\partial x} (u\sigma_x + v\tau_{xy} + w\tau_{xz}) + \frac{\partial}{\partial y} (u\tau_{yx} + v\sigma_y + w\tau_{yz}) + \frac{\partial}{\partial z} (u\tau_{zx} + v\tau_{zy} + w\sigma_z) \right] \tag{26}$$

The total energy per unit mass within the control volume is the sum of the internal and kinetic energies, given by;

$$Total\ Energy = e + \frac{V^2}{2}. \tag{27}$$

The variation in kinetic and internal energy for the control volume is shown in equations (28) and (29), respectively.

$$dE_{internal} = d(\rho e)dxdydz \tag{28}$$

13

$$dE_{kinetic} = d(\rho \frac{V^2}{2})dxdydz \qquad (29)$$

Rewriting these two equations and summing them, the variation of total energy can be written as;

$$\frac{dE}{dt} = \frac{\mathbf{D}}{\mathbf{D}t}(\rho e + \rho \frac{V^2}{2})dxdydz. \qquad (30)$$

Changes in heat content due to conduction only are considered. According to Fourier's Law the heat flux is proportional to the temperature gradient, so that the heat variation due to conduction can be written as;

$$\frac{1}{A}\frac{dQ}{dt} = q = -k\frac{\partial T}{\partial n}. \qquad (31)$$

Thus, by equating the amount of heat transferred into the volume with the amount of heat leaving the volume, the following relation is obtained;

$$-k\frac{\partial T}{\partial x}dydz + (k\frac{\partial T}{\partial x} k\frac{\partial T}{\partial x}dx)dydz$$

which gives the heat flux in the x-direction,

$$\frac{\partial}{\partial x}k\frac{\partial T}{\partial x}dxdydz. \qquad (32)$$

Repeating similar procedures for the y-direction and the z-direction a final expression for the total heat variation is given by;

$$\frac{dQ}{dt} = \left[\frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + \partial(k\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k\frac{\partial T}{\partial z})\right]dxdydz. \qquad (33)$$

Substituting equations (26), (30) and (33) back into equation (24), the general energy equation is obtained. [Ref. 3,19]

$$\frac{\mathbf{D}}{\mathbf{D}t}(\rho e + \rho \frac{V^2}{2}) = \frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k\frac{\partial T}{\partial z})$$

$$+ \frac{\partial}{\partial x}(u\sigma_x + v\tau_{xy} + w\tau_{xz}) + \frac{\partial}{\partial y}(u\tau_{yx} + v\sigma_y + w\tau_{yz}) + \frac{\partial}{\partial z}(u\tau_{zx} + v\tau_{zy} + w\sigma_z) \qquad (34)$$

14

B. CONSERVATION LAW FORMULATION

The primitive variable formulation of the Navier-Stokes equations shown in the previous section can be put into conservation law form using vector identities. The conservation law form can be also derived by applying conservation principles on a control volume. Because physical insight is gained by this procedure the conservation law form derivation is outlined in the next section.

This formulation stems from the fact that certain quantities (i.e. mass, momentum, energy) for a fluid in motion are conserved. Conservation implies that the flux of a quantity crossing a control surface and the net effect of internal sources results in a variation of the conserved quantity. These sources and fluxes depend on time and space as well as fluid motion. The fluxes are vectors for a scalar quantity and tensors for a vector quantity. Mass and energy are examples of a scalar quantity whereas momentum is a vector quantity. Molecular motion and convective transport of a fluid contribute to flux. Molecular motion has the tendency to make the fluid homogeneous and has a diffusive effect.

1. General Form of Conservation Law

a. Scalar Conservation Law

Considering a scalar quantity U within a control volume V, the time variation of the quantity U is;

$$\frac{\partial}{\partial t} \int_V U \, dV.$$

This should be equal to the incoming fluxes ($\vec{F} = U\vec{V}$) through a surface S (where $\vec{n}$ is the unit normal vector pointing outward),

$$-\int_S \vec{n} \cdot \vec{F} \, dS = -\int_S \vec{F} \cdot \vec{dS}$$

plus any possible contribution from sources of U.
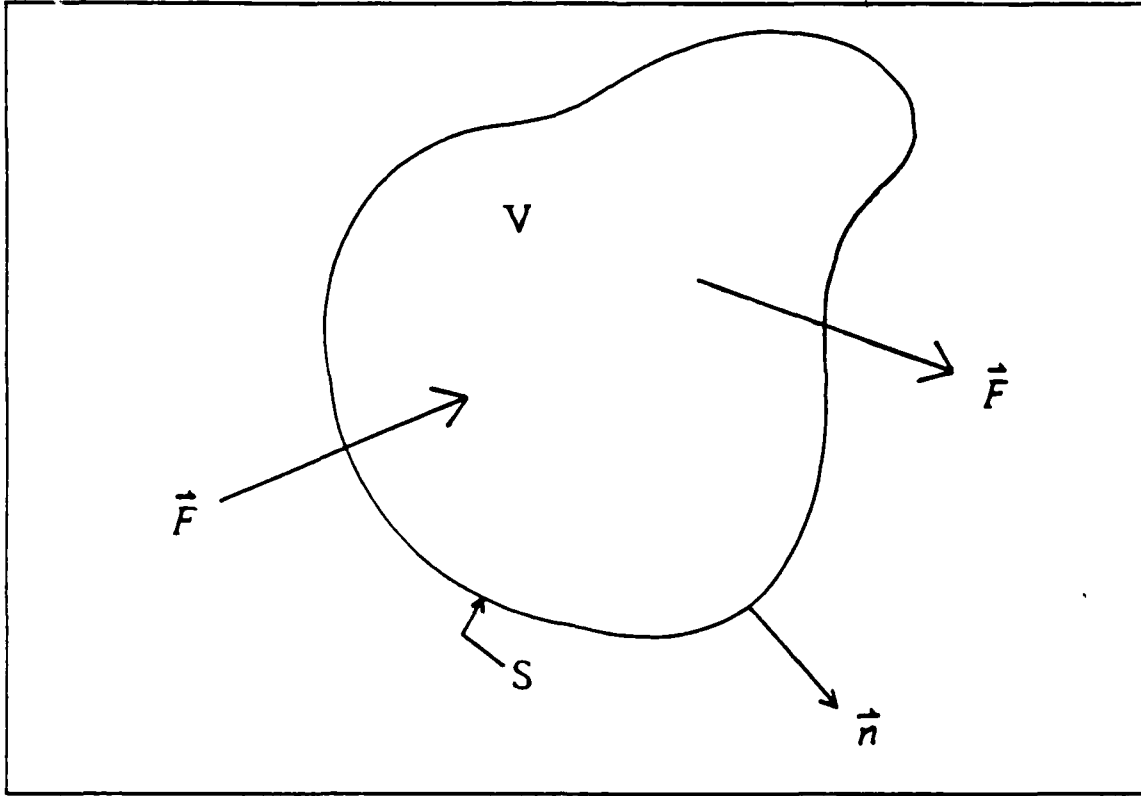
Figure 5.   Flux Diagram

The flux vector $\vec{F}$ has two components, a diffusive contribution and a convective contribution. The sources can be written as the addition of volume sources Q_V and surface sources $\vec{Q}_S$;

$$\int_V Q_V dV + \int_S \vec{Q}_S \cdot d\vec{S}$$

so that the final conservation equation for the scalar quantity U is,

$$\frac{\partial}{\partial t} \int_V U dV = \int_V Q_V dV + \int_S \vec{Q}_S \cdot d\vec{S} - \int_S \vec{F} \cdot d\vec{S}. \qquad (35)$$

When using Gauss's Theorem, equation (35) can be rewritten as;

$$\int_V \frac{\partial U}{\partial t} dV + \int_V \vec{\nabla} \cdot \vec{F} dV = \int_V Q_V dV + \int_V \vec{\nabla} \cdot \vec{Q}_S dV.$$

16

For an arbitrary volume V the differential form of the conservation law is given by equation (36).

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot (\vec{F} - \vec{Q}_S) = Q_V \qquad (36)$$

b. Vector Conservation Law

As stated earlier if the conserved quantity $\vec{U}$ is a vector, then the flux and surface source become tensors; $\vec{F} = \vec{U} \cdot \vec{V}$, $\vec{F}_s$, and the volume source in turn becomes a vector, $\vec{Q}_V$. An analogous derivation as in the conservation of a scalar quantity can be done for a vector quantity, whose integral and differential form are shown below.

$$\frac{\partial}{\partial t} \int_V \vec{U} dV + \int_S \vec{\vec{F}} \cdot \vec{dS} = \int_V \vec{Q}_V dV + \int_S \vec{\vec{Q}}_S \cdot \vec{dS} \qquad (37)$$

$$\frac{\partial \vec{U}}{\partial t} + \vec{\nabla} \cdot (\vec{\vec{F}} - \vec{\vec{Q}}_S) = \vec{Q}_V \qquad (38)$$

In equation (37) the convective component of the flux tensor can be written in tensor form as:

$$F_{C_{ij}} = v_i U_j$$

where v is the velocity vector. The diffusive component of the flux for a homogeneous system can be written as;

$$F_{D_{ij}} = -\rho \kappa \frac{\partial u_j}{\partial x_i}$$

where κ is the diffusivity constant. Equation (35) or (36) is the basic formulation of the conservation law for a general case. When continuity of flow properties is assumed (i.e. no shocks present), then equations (36) and (38) are valid. [Ref. 5: pp. 25-55]

2. Equation of Mass Conservation

In this particular instance the property U is mass and no diffusive flux is present, only convection. Therefore equation (35) can be directly written as;

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \vec{V} \cdot \vec{dS} = 0$$

or in differential form as in equation (39). [Ref. 5: p. 33]

17

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V}) = 0 \tag{39}$$

3. Equation of Momentum Conservation

For this case the conserved quantity is momentum which is a vector. From Newton's Second Law it was mentioned that change of momentum is due to external volume forces and internal forces. Assuming a Newtonian fluid, the stresses can be written as;

$$\vec{\sigma} = -p\vec{I} + \vec{\tau}$$

where $\vec{I}$ is the unit tensor, so that $-p\vec{I}$ is the hydrodynamic pressure along the diagonal. The $\vec{\tau}$ term is the viscous shear stress tensor, equation (15), which is written as;

$$\tau_{ij} = \mu((\partial_i v_j + \partial_j v_i) - \frac{2}{3}(\vec{\nabla} \cdot \vec{V})\delta_{ij}).$$

Referring to equation (37) and assuming that the external volume forces is zero the integral form for the conservation of momentum is;

$$\frac{\partial}{\partial t} \int_V \rho \vec{V} dV + \int_S \rho \vec{V} \cdot (\vec{V} \cdot \vec{dS}) = \int_S \vec{\sigma} \cdot \vec{dS}$$

and applying Gauss's Theorem,

$$\int_V \frac{\partial}{\partial t} \rho \vec{V} dV + \int_V \vec{\nabla} \cdot (\rho \vec{V} \otimes \vec{V}) dV = \int_V \vec{\nabla} \cdot \vec{\sigma} dV$$

where $\otimes$ indicates the tensor product of two vectors. This can be written in differential form as shown below. [Ref. 5: pp. 40-50.]

$$\frac{\partial}{\partial t}(\rho \vec{V}) + \vec{\nabla} \cdot (\rho \vec{V} \otimes \vec{V} + p\vec{I} - \vec{\tau}) = 0 \tag{40}$$

4. Equation of Energy Conservation

The quantity being conserved is energy, E and from the First Law of Thermodynamics the variation in energy must balance with the work of the forces acting on the system including any heat addition. The convective flux of energy can then be written as;

$$\vec{F}_C = \rho \vec{V} E$$

18

where E is the sum of the internal energy plus kinetic energy. Definition of the diffusive flux term describes that diffusion of heat for a fluid at rest is due to molecular thermal conduction, and using Fourier's Law of Heat Conduction, the diffusive flux term can be written as;

$$\vec{F}_D = k\vec{\nabla} \cdot \vec{V}T$$

where k is the thermal conductivity and T is the absolute temperature. Assuming no radiation, chemical reactions or work due to external forces, again the Q_v volume source is zero. The net work done on the fluid by the internal shear stresses acting on the surface of the control volume is given by;

$$\vec{Q}_S = \overset{=}{\sigma} \cdot \vec{V}.$$

Using equation (37) and substituting the quantities obtained above, the equation for energy conservation can be written as;

$$\frac{\partial}{\partial t}\int_V \rho E dV + \int_S \rho E\vec{V}\cdot d\vec{S} = \int_S kT\vec{\nabla}\cdot d\vec{S} + \int_S (\overset{=}{\sigma}\cdot\vec{V})\cdot d\vec{S}$$

or in differential form as in equation (41).

$$\frac{\partial}{\partial t}(\rho E) + \vec{\nabla}\cdot(\vec{V}(E\rho + p) - k\vec{\nabla}T - \overset{=}{\tau}\cdot\vec{V}) = 0 \qquad (41)$$

Equation (41) can be rewritten as;

$$\rho\frac{De}{Dt} + p\vec{\nabla}\cdot\vec{V} = \frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k\frac{\partial T}{\partial z}) + \mu\Phi$$

where Φ is called the dissipation function. Dissipation represents the heat equivalent of the rate at which the mechanical energy is lost during deformation of the medium due to viscosity. The dissipation function is given by;

$$\Phi = 2\left[(\frac{\partial u}{\partial x})^2 + (\frac{\partial v}{\partial y})^2 + (\frac{\partial w}{\partial z})^2\right] + (\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y})^2 +$$

$$(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z})^2 + (\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x})^2 - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z})^2.$$

19

The system of partial differential equations given by equations (39), (40) and (41) can be written in compact vector notation as;

$$\frac{\partial \vec{q}}{\partial t} + \vec{\nabla} \cdot \vec{Q} = 0 \qquad (42)$$

where $\vec{q}$ is the vector of dependent conservative variables and $\vec{Q}$ is a vector composed of the nonlinear inviscid and viscous fluxes.[Ref. 5: pp. 45-50]

5. Strong Conservation Form

The strong conservation law form given by equations (39), (40) and (41) in vector notation can be written for a Cartesian coordinate system as;

$$\frac{\partial \vec{q}}{\partial t} + \frac{\partial \vec{E}}{\partial x} + \frac{\partial \vec{F}}{\partial y} + \frac{\partial \vec{G}}{\partial z} = \frac{\partial \vec{R}}{\partial x} + \frac{\partial \vec{S}}{\partial y} + \frac{\partial \vec{T}}{\partial z} \qquad (43)$$

where $\vec{q}$ is the vector of conservative variables and $\vec{E}$, $\vec{F}$, and $\vec{G}$ are the flux vectors given by:

$$\vec{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \qquad (44)$$

$$\vec{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ (p + e)u \end{bmatrix} \quad \vec{F} = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v w \\ (p + e)v \end{bmatrix} \quad \vec{G} = \begin{bmatrix} \rho w \\ \rho w u \\ \rho w v \\ \rho w^2 + p \\ (p + e)w \end{bmatrix}. \qquad (45)$$

The vectors of $\vec{R}$, $\vec{S}$, and $\vec{T}$ contain the viscous terms. When they are omitted, the Euler equations are recovered.

$$\vec{R} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ (\vec{\vec{T}} \cdot \vec{V} + \vec{q}_c)_x \end{bmatrix} \quad \vec{S} = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ (\vec{\vec{T}} \cdot \vec{V} + \vec{q}_c)_y \end{bmatrix} \quad \vec{T} = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ (\vec{\vec{T}} \cdot \vec{V} + \vec{q}_c)_z \end{bmatrix} \tag{46}$$

The product term of $\vec{\vec{T}} \cdot \vec{V}$ is written in component form as below.

$$(\vec{\vec{T}} \cdot \vec{V})_x = \tau_{xx} u + \tau_{xy} v + \tau_{xz} w$$

$$(\vec{\vec{T}} \cdot \vec{V})_y = \tau_{yx} u + \tau_{yy} v + \tau_{yz} w \tag{47}$$

$$(\vec{\vec{T}} \cdot \vec{V})_z = \tau_{zx} u + \tau_{zy} v + \tau_{zz} w$$

The heat flux vector $\vec{q}_c$ is the heat transfer by conduction and can be written as;

$$\vec{q}_c = -k\nabla T = -K(a_x^2, a_y^2, a_z^2)^T \tag{48}$$

where,

$$K = \frac{\mu}{Pr(\gamma - 1)} \qquad Pr = \frac{c_p \mu}{k} .$$

In the above equations a denotes the speed of sound, Pr is the Prandtl number, c_p is the specific heat at a constant pressure and e is the total energy per unit volume. [Ref. 20] Pressure and energy are related by the perfect gas law as follows.

$$p = (\gamma - 1)\left[e - \frac{\rho}{2}(u^2 + v^2 + w^2) \right]$$

These equations can be transformed into different curvilinear coordinate systems in order to facilitate the numerical implementation.

A coordinate mapping is introduced which allows the transformation of the equations of motion from a Cartesian coordinate, time varying. nonorthogonal coordinate system. The mapping is linked to the Cartesian coordinates as follows;

$$\xi = \xi(x,y,z,t)$$

21

$$\eta = \eta(x,y,z,t)$$

$$\zeta = \zeta(x,y,z,t).$$

The Cartesian coordinate system is the physical domain, and the transformed space is referred to as the computational domain. This computational domain is orthogonal with a uniform rectangular mesh so that unweighted differences can be taken to form the derivatives.

The thin layer compressible Navier-Stokes equations are obtained from equation (43) by retaining the viscous terms only along the direction that is normal to the body. Also, the derivatives of the stress terms in the crossflow (i.e. y, z) directions are discarded. The thin layer formulation of the strong conservation law form of the governing equations for a curvilinear coordinate system (ξ, η, ζ) along the axial, circumferential, and normal direction, respectively can be written as;

$$\frac{\partial \hat{q}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{H}}{\partial \zeta} = \frac{1}{Re} \frac{\partial \hat{S}}{\partial \zeta} \tag{49}$$

where $\hat{q}, \hat{F}, \hat{G}, \hat{H}$, and $\hat{S}$ are,

$$\hat{q} = \frac{1}{J} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \qquad \hat{F} = \frac{1}{J} \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (e+p)U - \xi_t p \end{bmatrix}$$

$$\hat{G} = \frac{1}{J} \begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (e+p)V - \eta_t p \end{bmatrix} \qquad \hat{H} = \frac{1}{J} \begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (e+p)W - \zeta_t p \end{bmatrix}$$

22

$$
\hat{S} = \frac{1}{J}
\begin{bmatrix}
0 \\
\mu m_1 u_\zeta + (\mu/3) m_2 \zeta_x \\
\mu m_1 v_\zeta + (\mu/3) m_2 \zeta_y \\
\mu m_1 w_\zeta + (\mu/3) m_2 \zeta_z \\
\mu m_1 m_3 + (\mu/3)m + 2(\zeta_x u + \zeta_y v + \zeta_z w)
\end{bmatrix}.
$$

Furthermore, it is defined that;

$$
m_1 = \zeta_x^2 + \zeta_y^2 + \zeta_z^2
$$

$$
m_2 = \zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta
$$

$$
m_3 = (u^2 + v^2 + w^2)/2 + \frac{\kappa}{Pr}\left(\frac{\partial a^2}{\partial \zeta}\right)
$$

and U, V, and W are the contravariant velocity components given by,

$$
U = u\xi_x + v\xi_y + w\xi_z + \xi_t
$$

$$
V = u\eta_x + v\eta_y + w\eta_z + \eta_t
$$

$$
W = u\zeta_x + v\zeta_y + w\zeta_z + \zeta_t.
$$

Again analogous to the previous derivations the pressure is related to density and total energy through the equation of state for an ideal gas.

C. NUMERICAL IMPLEMENTATION

1. The Numerical Algorithm

The solutions over a strake-delta wing configuration resembling a modern fighter aircraft planform will be presented in the last part of this thesis. Even though the main effort of this work was not the numerical solution of the governing equations (i.e. the compressible Navier-Stokes equations), the technique used for the numerical implementation is briefly described in the following paragraphs.

The numerical scheme used for the solution of the governing equations is based on a finite difference discretization of the thin layer Navier-Stokes equation [Ref. 6]. The numerical integration was performed using a partially flux-split numerical scheme. Upwinding was performed in the main flow direction using flux vector splitting while

23

central differencing was used in the other two directions. The factored form of the resulting algorithm is as follows;

$$\left[I + h\delta_\zeta^b (A^+)^n + h\delta_\zeta C^n - hRe^{-1}\overline{\delta}_\zeta J^{-1} M^n J - D_{i|\zeta} \right]$$

$$\times \left[I + h\delta_\zeta^f (A^-)^n + h\delta_\eta B^n - D_{i|\eta} \right] \Delta q^n =$$

$$-\Delta t (\delta_\zeta^b \left[(F^+)^n - F_\infty^n \right] + \delta_\zeta^f \left[(F^-)^n - F_\infty^n \right] + \delta_\eta (G^n - G_\infty) + \delta_\zeta (H_n - H_\infty) + \frac{1}{Re}\delta_\zeta (S^n - S_\infty))$$

$$- D_e(q^n - q_\infty).$$

The explicit dissipation D_e was used along the directions where central differencing was applied. The implicit dissipation term D_i was added for numerical stability. Steady aerodynamic flows at subsonic flows ($M = 0.2$) do not contain shock waves and can be quite well predicted by a central difference scheme that is augmented by these dissipation terms.[Ref. 20]

2. Turbulence Model

Simulation of high Reynolds number flows is obtained by the solution of the Reynolds averaged Navier-Stokes equations. These equations have extra unknowns and are commonly called the Reynolds stresses [Ref. 3]. The relations between the Reynolds stresses and the mean flow quantities is the well known closure problem. In practice some turbulence model is used which relates the Reynolds stresses with the mean flow quantities. The turbulence model selected for this research was an algebraic eddy viscosity. This model is the Baldwin-Lomax model as modified by Degani and Schiff to treat three-dimensional separated flows [Ref. 21,22].

The turbulence is simulated in terms of an eddy viscosity coefficient μ_t. The coefficient of viscosity and the heat flux term in the Navier-Stokes equations are replaced with $\mu + \mu_t$ and $\frac{\mu}{p_r} + \frac{\mu_t}{p_{r_t}}$, respectively. The turbulence model is similar to one developed by Cebeci with modifications that allow for the locating of the boundary layer [Ref 23]. A two layer algebraic eddy viscosity model is used where the Prandtl-Van Driest formulation is used in the inner region and the Clauser formulation is used in the outer region [Ref. 21]. The inner region is any normal distance from the wall, y, that is less than or equal to $y_{crossover}$. If this is the case then μ_t is defined by the following expression;

$$(\mu_t)_{inner} = \rho l^2 | \omega |$$

24

where,

$$l = ky \left[1 - \exp(\frac{-y^+}{A^+}) \right]$$

and,

$$| \omega | = \sqrt{(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y})^2 + (\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z})^2}$$

$$y^+ = \frac{\rho_w u_z y}{\mu_w} = \frac{\sqrt{\rho_w \tau_w} \cdot y}{\mu_w} .$$

If y is greater than $y_{crossover}$ then μ_t is defined by;

$$(\mu_t)_{outer} = K C_{Cp} F_{Wake} F_{Kleb}(y)$$

where K is the Clauser constant, C_{C_p} is an additional constant, and for boundary layers,

$$F_{Wake} = y_{max} F_{max}$$

or for wakes and separated boundary layers,

$$F_{Wake} = C_{Wk} y_{max} \frac{U_{Dif}^2}{F_{max}} .$$

In the above equation U_{Dif} is the difference between the maximum velocity at y_{max} and the minimum velocity in the profile. The quantities of y_{max} and F_{max} are calculated using;

$$F(y) = y | \omega | \left[1 - \exp(\frac{-y^+}{A^+}) \right].$$

The function $F_{Kleb}(y)$ is the Klebanoff intermittency factor and is defined as;

$$F_{Kleb}(y) = \left[1 + 5.5(\frac{C_{Kleb} y}{y_{max}})^6 \right]^{-1} .$$

All other parameters are constants determined empirically and given in [Ref 21]. The use of this model eliminates the need for finding the edge of the boundary layer and reduces one of the sources of error in the Navier-Stokes solutions.

III. SURFACE GRID GENERATION

The focal point of this work was the generation of the computational mesh over a three-dimensional strake-delta wing configuration that models a modern fighter aircraft planform. Therefore, in the next two chapters the surface and field grid generation procedures are described. The considerations which must be taken into account in order to construct a surface grid suitable for the subsequent generation of the field are discussed. Finally, the various field grid generation methods are discussed and two different approaches which were used to generate the field grid over the strake-delta wing configuration are described.

The first step to be taken in establishing a finite difference or finite element scheme for solving a system of partial differential equations is to replace the continuous domain by a finite mesh, commonly known as a grid. Grid generation is one of the central problems in the procedure to obtain a numerical solution. A well constructed grid greatly facilitates the numerical solution of a system of P.D.E.s. On the other hand, an improper grid choice may lead to instabilities, inaccuracies and/or lack of convergence. Numerical grid generation is a procedure for the orderly distribution of observers over the physical field domain in such a way that efficient communication among the observers is possible. Also, it assures that all physical phenomena of interest in the entire field may be represented with sufficient accuracy by this finite collection of observers.

Grid generation for two-dimensional domains is relatively simple and may be achieved with purely algebraic techniques, even for relatively complex domains [Ref. 24]. In addition, for suitable geometries of the boundaries conformal mapping techniques may be used. Conformal mapping techniques have the advantage that they are relatively simple and inexpensive [Ref. 25 : pp. 488-490][Ref. 4: pp. 7-56]. They also preserve grid orthogonality, but their use is limited to domains with simple boundaries where a conformal transformation between the physical domain and a simpler transformed domain may be readily defined.

The generation of a computational grid for a three-dimensional domain, however, presents greater difficulties. For a limited class of external and internal domains it is sometimes possible to fill the entire three dimensional domain with a sequence of two-dimensional plane grids that will constitute the entire three-dimensional field grid. An application of this idea is shown later for the construction of the field grid over the

double-delta wing. In many instances however, it is either difficult to decompose the three-dimensional domain into a sequence of two-dimensional domains, or it is preferable to construct a purely three-dimensional mesh. Of course, both the complexity and computing time of a three-dimensional grid generation method will be higher. In any case, the definition of the surface boundaries must be done precisely and accurately.

Before any work can be started on a three-dimensional field grid over a body one must first define the surface geometry of the body. The definition of the body's surface and its quality is imperative to the success of the field grid. There exist many avenues to create the surface grid, some include algebraic techniques, cubic Hermite functions, Bezier curves or Non-Uniform Rational B-splines (NURBS) [Ref. 4: pp. 237-249] [Ref. 25: pp. 497-503]. Each of these techniques has its advantages and the exact method that will best fit a particular surface will vary. The availability of accurate data for description of the surface geometry will also play a major role in the generation of the surface grid. If all surfaces can be defined in terms of equations, then an algebraic technique might prove to be the most efficient. Whereas, if the surface is very complex, as is the case for actual aircraft surfaces, all that is available are two-dimensional cross-sections and a curve fitting technique will have to be used. Whatever the method, it is of utmost importance that the surface grid generation program be written in such a way that it will enable maximum flexibility in the number of grid points and their distribution. This early concern and respect for versatility will pay large dividends upon subsequent generation of the field grid. For even after the surface grid has been completed, an interactive trial and error process of changing the surface grid will be required to achieve an effective field grid.

A. DOUBLE-DELTA WING SURFACE GRID

The dimensions of the double-delta wing model are shown in Figure (7). From this figure it can be easily seen that most of the surfaces can be defined by linear relationships with the only exception being the NACA 64-005 cross-section. For this reason algebraic grid generation on the surface was chosen. For a linear relationship and analytically defined points no advantage is gained by using a curve fitting method. The part of the wing that contains the NACA 64-005 airfoil cross-section required special treatment. Generation of the surface grid over the airfoil cross-section would require a curve fitting technique. Much work has been done in this area and NURBS can produce excellent results in approximating airfoils. The main advantage of this technique is that it provides the flexibility of modifying the cross-section or shape of the surface by simple

27

changes of user specified parameters. The disadvantage is that the complexity is higher and the redistribution of the points that represent the airfoil is also more difficult. Because the cross-section is a NACA airfoil whose contour shape can be well approximated by straight lines, the use of a purely algebraic technique for the entire double-delta wing surface grid was utilized.

Once this decision was made, the areas containing singularities had to be identified and the program for the algebraic grid generation had to be written. Because the surface grid is used as initial or boundary conditions for the generation of the field grid, much care had to be taken to avoid any singularities that would propagate into the field grid. In addition, special care must be taken at the regions of sharp corners such as the leading edge. In these areas it is not possible to maintain field grid orthogonality and the location of these acute angles can be seen in Figure (7). In general, these areas are found on the entire leading edge, at the apex and at the rectangular edge near the wingtip. These corners had to be approximated by "rounding" off these areas with a radius that was very small. The radius used was 0.001% of the chord length, so to the naked eye the surface grid appears to be a sharp corner. This rounding of acute angles allows the field grid to maintain orthogonality which is a desirable feature for subsequent numerical implementation; see Figures (11), (14), (17), (20) and (22). Also, high grid resolution is provided at the same time in these areas where the change of the flow field variables is expected to be rapid. The methodology for generating the source code that would compute the grid points was to progress from the nose in an axial direction through to the wake. The grid points were essentially generated for a two-dimensional cross-section in the yz-plane, then an incremental step in the x-direction was made and again the grid points for a new yz-plane were computed. The source code was written in five logical sections that defined regions of the wing with similar cross-sections. These five sections were the apex, the strake, the wing, the trailing edge rectangular section and the wake.

1. **The Apex**

Special care had to be taken in the modeling of the nose region. The apex of the wing is a single point that transitions to a diamond shape cross-section. Taking into account that smoothness has to be maintained, a hemisphere may be used to provide smooth transition between the singular point of the apex and the diamond cross-sections at the nose, see Figure (11). The radius of this hemisphere is 0.001% of the chord length and allows for a smooth transition. The radius of the sphere, the number of grid points for the axial (x-direction) and the circumferential (y-direction) were inputted by the user. An incremental angle was determined for both the xz-planes and yz-planes which then

28

enabled the computation of the grid points. The yz-plane cross-sectional grid points were then calculated using incremental yz-plane angles as the x-location progressed downstream using the incremental xz-plane angles.

2. The Strake

The main concern in defining this section was that the leading edge has a corner that is relatively sharp and would preclude a field grid that is orthogonal. Therefore, the sharp leading edge was approximately rounded as shown in Figure (12) through Figure (17). The computations here involved a user specified radius that was the same as the one in the approximation of the apex. This radius was maintained to allow a smooth transition from the sphere of the apex to the diamond cross-section of the strake. The surface grid generator code provides the versatility to change the number of grid lines in this radius which is kept constant for every cross-section. This issue becomes important as the ratio of radius to wing span drastically changes between the apex and junction of the strake and wing. Again similar logic to the one used to define the apex was used here. The difference being that the increment of the x-coordinate was computed depending on the number of grid lines along the x-direction used to define the strake part of the body. Simple relations from analytic geometry are used to determine the y and z-coordinate as a function of the x-location.

The distribution or clustering of the grid lines will be discussed in more detail later in this section. It is important to mention that the distances between successive grid points, along the x and y-directions was determined by calling a subroutine. This allows to experiment with many different distributions with a simple change of input parameters.

3. The Wing

For the wing section three areas required special attention. The first was like the strake, in that the leading edge forms a corner which unlike the strake was not as sharp. This was because of the NACA 64-005 cross-section has a finite curvature at the leading edge. Part of the leading edge did not require rounding and the number of grid lines approximating the edge was reduced, see Figure (15) through Figure (17). The NACA 64-005 cross-section is generated by a subroutine which requires as input only the normalized root chord length of the airfoil, $x_c = x_c(y)$. The area of the wing spanning between the wing centerline and the part of the wing having a NACA 64-005 cross-section was defined by linear interpolation. Some sort of curve fitting method could have been used, but since the wing is thin and the distance is short, a linear approximation was assumed to be sufficiently accurate.

29

4. The Trailing Edge Rectangular Section

This particular section required the most challenging surface grid definition in the early stages. This was primarily because the wingtip has a variable finite thickness that depends on the x-location, see Figure (18) through Figure (20). The actual calculation of the grid point locations was simple, but the number of grid lines at the wingtip region had to change due to this variable thickness. This required to take grid lines out of the edge and redistribute them back onto the upper and lower surfaces. This is the reason why the grid lines in this section, when viewed in the xy-plane appear staggered, see Figure (9). The trailing edge has a finite thickness (0.002% of chord length) and thus avoids any singularities that unnecessarily complicate the subsequent numerical implementation and the generation of the field grid.

5. The Wake

For the numerical implementation, an extension of the far end of the computational domain of 2.0 - 3.0 root chords beyond the body is required. The wake was relatively simple to generate because all that varied was the x-coordinate. All the yz-planes remain constant from the trailing edge to the end of the grid. Examples of this cross-section can be seen in Figure (21) and Figure (22). The wake extends for 2.0 root chord lengths beyond the wing trailing edge. This length was selected because it allowed for a smooth transition from the wing to the wake for a given number of x-direction grid lines.

B. DISTRIBUTION PARAMETERS

Flexibility in the distribution of the surface grid points in both the x-direction and the y-direction, which are shown in Figure (10), is important for the surface grid. The foremost problem is to ensure that the distance between successive grid points makes a smooth transition. Of course, the spacing between the surface grid points could have been made the same, but this is impractical because the total number of grid lines would be excessive due to the small radius that was used to approximate the corners. Therefore a distribution of grid points must be developed that is very dense at the corners and sparser in the other regions. High grid clustering is also required in areas where steep gradients in the flow-field are expected, such as the leading edge where the leading edge vortices appear. The distribution in the y-direction can be seen in Figure (9) and Figure (10) for various cross-sections of the wing and the distribution for the x-direction can be seen in Figure (10). The y-direction has a high grid clustering around the leading edge which becomes sparser near the centerline of the wing. This was the general procedure

followed for the y-direction distributions in all cross-sections. The distribution along the x-direction required a higher density in the nose region and sparser distribution at the area near the end of the wake. A subtle change to a higher density occurs where the wing has a geometry change as can be seen in Figure (9).

Stretching of the grid points along a coordinate direction can be obtained by using simple algebraic functions such as linear, exponential mapping or trigonometric functions. Use of these functions allows for a smooth transition from sparse grid densities to high grid densities. A quadratic function was first attempted but this resulted in a distribution that became less dense too fast and would spread the grid points out to an excessive amount near the centerline. Next a linear stretching function was used and this gave much better results but did not allow a "smooth" transition from the high grid density region to the region with sparser grids. This effect was more pronounced along the y-direction. The linear function allows a more constant distribution over the whole wing in the x-direction. The linear equation shown below was used for the linear stretching:

$$x_{j+1} = cx_j$$

where the user specifies the parameter c depending on the desired degree of stretching. A value of $c = 1.0$ would result in an equidistance spacing, whereas a $c = 2.0$ would result in a high clustering of grid points near one or both ends. Difficulties were not encountered in the x-direction because the transition from the low to high density distributions were not as extreme as in the y-direction. By changing the linear stretching parameter, a smooth transition from more to less dense areas was achieved. The grid stretching in the axial direction required different values of c depending on the wing section; for example, $c = 1.005$ was used prior to the trailing edge and $c = 1.205$ after the trailing edge of the wing. The effect of these constants can be seen in Figure (9). Each representative cross-section had to be investigated and a constant assigned that allowed a smooth transition from one section to another. These constants were determined by a trial and error procedure, but experience gained by many iterations expedited the process. From the many iterations for the surface grid alone, an appreciation for the flexibility of the source code was gained.

To resolve the y-direction distribution, the stretching function first attempted was exponential which proved to be inadequate. The exponential stretching is obtained by using the following expression;

31

$$y_{k+1} = cy_k^s$$

where c and s are parameters chosen by the user to produce a desired distribution. The dimensions of the radius used to approximate corners are so small compared to the characteristic dimension (i.e. the chord length) that for the desired number of grid points in the y-direction the transition did not occur smoothly. Finally, a sinusoidal distribution produced better results. The equation below was used to obtain this distribution;

$$y_k = c \sin b$$

here c is a user specified parameter and b is allowed to incrementally change over a specified range of angles in order to obtain the desired section of the sine curve. The distribution produced the best results for a sine curve segment from 0 to 45 degrees. Freedom was written into the source code to use constants to finely adjust the distribution but were not required because of sufficient results without them.

C. PROGRAM FEATURES FOR THE SURFACE GRID

The source code for the surface grid can be seen in Appendix E. The main concern in the construction of the source code for this problem was to give the author maximum flexibility in the generation of this surface grid. Listed below are some of the features that can be easily changed via an input file.

- The number of grid points in the x-direction at five different sections.
- The number of total grid points in the y-direction.
- The radius used in the approximations of sharp edges.
- The number of grid points used in the radius for the corner approximations.
- The sweep angles of the strake and the wing.
- Maximum widths of the strake and wing.
- Lengths of the strake, the wing, the rectangular section and the wake.
- Distribution constants at five locations in the x-direction and at six locations in the y-direction.

This flexibility in the surface grid paid a major dividend in the subsequent generation of the field grid. This is because the surface and field grid generation is an interactive process that usually requires changes in the surface grid. Another feature is that the distribution functions are written as subroutines. This allowed the author the flexibility of trying different functions based on the geometry and desired gradients of distribution.

32

Another subroutine is the calculation of the grid points that are part of the NACA 64-005 airfoil cross-section. This enabled the changing of this cross-section by merely changing two lines of the data statement. The subroutines that are included in Appendix E are linear, quadratic, exponential and sinusoidal. Again it is emphasized that the source code for the surface grid generation must be as versatile as possible.

This program can be used to generate a surface grid for a wing with dimensions that are different from the one chosen by the author. But because great care needs to be taken in grid point distribution a change in the dimensions would most definitely require an adjustment of radius, the number of points, distribution constants and even distribution functions. A close examination of every grid constructed, either visually or computationally, must be completed to ensure that no intersection of the grid lines occurs. The program that was originally written was continually revised to permit an adequate construction of the field grids. Included in Appendix F is the final program that was utilized for the generation of the surface grid for the final spherical field grid topology.

IV. FIELD GRID GENERATION

In this chapter different numerical techniques for the field grid generation are presented. Their advantages and disadvantages as far as grid generation and numerical implementation are explained. Most available field generation techniques require as input a surface grid which must be constructed before the field grid generation. The generation of the field grid is directly dependent on the distribution of the grid points on the body surface. There are several ways to generate field grids, a few of which include methods involving solutions of Elliptic Partial Differential Equations (P.D.E.), Parabolic P.D.E.s, or Hyperbolic P.D.E.s. For a limited class of problems algebraic methods can also be used. In the following paragraphs the various grid generation methods based on the solutions of P.D.E.s will be described. The hyperbolic method, which was used to generate the field grid over the double-delta wing will be explained in detail, whereas only a brief description of elliptic and parabolic techniques will be given.

The classification of P.D.E.s into hyperbolic, parabolic or elliptic type is obtained from the general form of the quasi-linear second order P.D.E., given by:

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu = g \qquad (1)$$

here $u = u(x,y)$ is the dependent variable and the coefficients a,b,c,d,e,f and g are functions of x and y . The type of equation (1) is determined from the sign of the quantity $b^2 - 4ac$ as follows.

$$b^2 - 4ac < 0 \quad (Hyperbolic) \qquad (2)$$

$$b^2 - 4ac = 0 \quad (Parabolic) \qquad (3)$$

$$b^2 - 4ac > 0 \quad (Elliptic) \qquad (4)$$

Each type of equation, hyperbolic, parabolic or elliptic has certain characteristic properties which can be successfully utilized for the grid generation in two and three-dimensional domains [Ref. 4: pp. 188-277]. For example, the solution of an elliptic equation in the interior of a domain depends on the specification of data over the entire boundary. Therefore, when a grid is generated by the solution of an elliptic P.D.E. all the boundary data must be specified.

The main feature of a hyperbolic type of problem is that the solution starts from an initial condition and propagates in time along certain directions known as the characteristic directions. Utilization of this property allows the construction of grids over contoured lines or surfaces by propagating in space the initial information provided by these lines or surfaces. The grid generation method must be carefully chosen to facilitate the type of grid desired. Each type of grid generation method will require some additional data to allow for a suitable solution. To some extent this additional data may determine which type of grid method should be utilized. If x and y are spatial coordinates (which is true for the present case) then the additional data will be the boundary conditions. If x and y represent time then the additional data will represent initial conditions.

A. ELLIPTIC GRID GENERATION

Field grid generation obtained by the solution of an elliptic set of equations requires specification of each and every boundary point of the closed domain where the grid will be generated. The inner boundary is simply the body surface grid and the outer boundary is a user specified shape. The body surface must be specified exactly. However, there is some flexibility in choosing the shape of the outer boundary. Another requirement of this method is that the curvilinear coordinates must be constant or monotonic on the boundaries. If any extrema of the curvilinear coordinates exist in the interior of the physical region then overlapping of the grid lines will occur. When using an elliptic method, initial boundary slope discontinuities are not propagated into the field. This feature of elliptic grid generators tends to make the grid very smooth. The large computational time requirement for the solution of the elliptic system of P.D.E.s can be a disadvantage. The simplest form of an elliptic P.D.E. is the Laplace equation;

$$\nabla^2 \xi^i = 0 \qquad (5)$$

where $i = 1,2$ for two-dimensional grid generation and $i = 1,2,3$ for three-dimensional grid generation. The effect of the Laplace operator is that a very smooth grid is produced which becomes equally spaced away from the boundary. The Laplacian also guarantees one to one mapping of the coordinate system. This method will have the effect of making the grid lines more closely spaced over concave boundaries and sparse over convex boundaries.[Ref. 4 : pp. 188-228][Ref. 25: pp. 503-510]

Another approach to generate the field grid is to solve a Poisson system of equations. This system has the following general form;

35

$$\nabla^2 \xi^l = P^l. \tag{6}$$

The forcing term P^l can be used to control the spacing and orientation of the grid lines. This control can be extended to move the intersection slope of the grid line with the boundary. When $P^l \to 0$ the grid lines tend to become equally spaced, i.e. to approach a grid obtained from the solution of Laplace's equation. The forcing term P^l can also be used to enhance grid orthogonality. Orthogonality of the grid lines close to the body surface grid does not occur normally in elliptical solutions. The main advantage of a Poisson type grid generator is that orthogonality control of the grid lines can be maintained at the expense of complex, lengthy and expensive calculations. There are several elliptic grid generators in use today that maintain grid orthogonality.[Ref. 4: pp. 193-236.] [Ref. 26]

B. HYPERBOLIC GRID

The hyperbolic method involves marching in space in a time-like fashion of the boundary information, i.e., a surface grid. This method is suitable for external flow problems where the exact location and shape of the outer boundary is of no vital importance. One major advantage is that computationally this method is efficient; in addition, orthogonality of the field grid is preserved. Hyperbolic methods are usually one to two orders of magnitude faster than the elliptic methods because of their noniterative time-like marching nature. Control of the grid lines is somewhat restrictive but specification of the cell volume can result in the avoidance of overlapping grid lines, especially in concave areas. Overlapping of the grid lines is not allowed because singularities are propagated into the field, so great care must be taken to avoid these when constructing the surface grid. Because the characteristics of the hyperbolic method include orthogonality preservation and computational efficiency, a hyperbolic grid generation method was selected as opposed to an elliptic grid generation method. Great care was taken in the generation of the surface grid to remove any singularities such as sharp corners, because rapid transitions on the surface geometry usually produce intersecting grid lines of the field grid. The wing configuration did not contain any severe concave surfaces which would cause intersecting of the grid lines. Two different grid topologies were examined for the grid generation of the field grid over a double-delta wing, i.e., cylindrical and spherical. First the cylindrical grid generation procedure is presented. [Ref. 4: pp. 272-276][Ref. 25: p. 503]

1. Cylindrical Grid Generation

The cylindrical type grid produces an H-O configuration shown in Figure (23). The grid in this figure is called H-O because when the wing is viewed from above or from the side the grid appears to have a "H" shape. When the grid is viewed from a nose-on viewpoint (i.e. the yz-plane), then the grid has an "O" shape, see Figure (24). It is usually easier to generate a cylindrical grid than generating a spherical grid because the three-dimensional cylindrical grid is an assembly of planar two-dimensional sections. This combination of two-dimensional planes starts at the nose and progresses aft towards the trailing edge, see Figure (25) and Figure (26). Various methods may be used to generate these two-dimensional grids on the planar cross-sections. Here, a two-dimensional hyperbolic grid generator was used to generate the plane O-type grids at various locations along the axial direction. The cylindrical grid has a singular point at the apex, Figure (25), where special care must be taken during numerical solution for the computation of the transformation metrics and the application of boundary conditions. Along the singular line starting from the apex and extending upstream to the beginning of the domain all the grid points collapse on to a single point.

Before the surface grid data points generated for a generic surface can be used for a cylindrical grid, a modification was required at the nose of the grid. As can be seen in Figure (7) a singularity is present at the very first point at the apex. To avoid the problems that this point can cause, the singular point at the apex was omitted. The field grid was then generated to have an annular type appearance in the yz-plane cross-section, see Figure (29). Because the radius is so small, the effects due to the inaccuracy of its surface definition are negligible.

The process of improving the quality of the field grid was completed using the computer graphics program PLOT3D [Ref. 27]. This graphics package was designed to facilitate visualization of the field and surface grids and flow fields of computed and experimental results. This same program was utilized to correct the surface grid during its development process. Because of the early concern regarding the surface grid and knowledge of hyperbolic grid shortcomings, only minor adjustments were required in the surface grid. These adjustments required a redistribution of surface grid points in the spanwise direction.

The initial surface grid had a linear distribution that was deemed inadequate just by visual inspection and from physical considerations of the flow field character at the leading edge region. A quadratic distribution was subsequently used which appeared to yield a better distribution of the grid points. Not until the field grid was generated was

37

it clear that this distribution was still insufficient. The necessary changes in the field grid were to smooth out the distribution near the leading edges and to increase the grid density where vortices were most likely to occur. Finally, a sinusoidal distribution along the spanwise direction gave the best results. Examples of various cross-sections of the surface grid distribution for the wing can be seen in Figures (29), (31), (34) and (37). At the trailing edge the same reasoning as for the nose was carried out and the grid was not collapsed onto a single line but instead retained a finite thickness. As a result a very small but finite thickness wake was generated.

The field grid was completed by extending the grid from the wing apex to a location 2.0 - 3.0 chord lengths upstream, where the freestream conditions can be applied. The reason for this is that the flow field is affected upstream by the presence of the wing. This addition to the grid was completed by repeating the very first annular shaped grid and by changing only the x-location. The yz-plane remained the same and it was repeated as the x-distribution gradually increased its Δx spacing until it reached a user specified x-location upstream. The short program for this additional grid can be seen in Appendix F and examples of the entire field grid (cylindrical type) can be seen in Figure (24) through Figure (41), Appendix B. The completed cylindrical field grid dimensions are 110x240x68.

The purpose of the grid generation is to facilitate a numerical solution to a system of P.D.E.s, and for accurate solutions to occur certain areas require special treatment. A problem that occurs in the computation of derivatives in the apex region is that differences are taken between points that may have different flow characteristics; i.e. one point may be in the boundary layer and the next point may be outside of this regime. This is the reason why clustering of grid points near the apex is required to avoid as much as possible these inaccuracies. Clustering of the grid points normal to the surface of the wing is naturally required to resolve the velocity gradients in the viscous boundary layer. For the most part though, the grid is aligned with the main flow direction and a numerical scheme that uses upwinding can produce accurate results.[Ref. 20]

2. Spherical Grid

This particular method of grid generation produces a C-O type grid as can be seen in Figure (24). This method is more complex than the cylindrical one because the entire three-dimensional grid must be generated simultaneously. The spherical grid topology has one singularity that is located at the apex and propagates upstream, see Figure (42) and Figure (43). The spherical grid is also aligned with the main flow; therefore, an upwinding scheme can be used for flow field solutions. One disadvantage

38

of the spherical grid topology is that the visualization of the flow is a little more difficult than for the cylindrical grid because the $\zeta = constant$ grid surfaces are not planes but three-dimensional surfaces. The flow field solution tends to converge faster on a spherical grid than in the cylindrical grid.

The hyperbolic grid generator for this work utilized a cell volume hyperbolic grid generation scheme. A coordinate transformation to the computational domain (ξ, η, ζ) was performed where the body surface was the boundary condition $\zeta(x, y, z) = 0$. The field grid is obtained by the solution of the following nonlinear system of P.D.E.s;

$$x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta = 0$$

$$x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta = 0$$

$$x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi = \Delta V.$$

where the initial condition at $\zeta = 0$ are the x, y, and z coordinates of the body surface [Ref. 20]. The first two equations are the relations that preserve orthogonality with respect to an outward normal vector ζ. The third equation is a user specified volume parameter that controls the cell size and normal spacing of the grid points. The grid is generated by "marching" in the ζ direction and the system of P.D.E.s is solved by an approximate-factorization, noniterative, implicit finite difference scheme. Even though grid orthogonality and smoothness are maintained the quality of the field grid is quite sensitive to the quality of the surface grid. Control of grid clustering along the normal to the surface direction is provided, but there is no accurate control in the location of the outer boundary due to the marching type solution. The outer boundary for the purposes of the present work is not crucial as long as it extends 2.0 - 2.5 root chord lengths away from the body.

The three-dimensional hyperbolic grid generation is very sensitive to the surface grid definition, since the surface grid distribution is propagated in space to generate the three-dimensional mesh. This sensitivity to the initial conditions is the reason why the spherical grid generation presented more difficulties than the cylindrical grid generation. The apex singularity in conjunction with the sharp angles created most of the problems in the grid generation as far as preservation of orthogonality is concerned. Because of this point singularity, a blunting of the nose was first attempted. This nose region also resulted in a reduction of grid lines in the x-direction to 110. For the C-O configuration the first grid line extends upstream and therefore it is not necessary to add axial grid lines

39

as in the cylindrical grid case. For effective field grid generation, distributions in all three directions had to be adjusted near the nose. After many iterations a solution was generated that required only minor adjustments. These adjustments were made by writing a small program that would algebraically adjust the x-distribution at the nose singularity. Algebraically fixed were the second and third grid planes in the x-direction. This technique is sometimes the only option available when the grid requires minor adjustments and the changing of usual parameters produces negative results. Although a suitable grid was constructed, an alternative method of allowing the apex to collapse to a sharp point was attempted in hopes of an even more suitable field grid. Also at this time solutions were being generated for the cylindrical grid and it was observed that the field grid required a higher grid density in the wing area to facilitate better definition of the vortex that develops over it. For this reason the number of grid lines in the axial direction was increased to 160. By allowing the apex to converge linearly to a point resulted in a three-dimensional mesh that had been unsurpassed up to this point. The disadvantage of allowing the strake to linearly collapse to a point was the original y-direction sinusoidal distribution (0° to 45°) of the surface grid was now insufficient. A solution to this problem was to allow the y-direction distribution near the apex to have a 45° to 60° distribution. This distribution results in a spacing of the grid lines that is nearly linear. Then the distribution was allowed to change incrementally as the x-location changed to achieve a 45° to 90° sinusoidal distribution at the strake and wing junction. This change was only required in the y-direction. The conscious decision to allow a singularity at the apex did not affect the quality of the field grid because the topology of the spherical C-O type grid requires that this area collapse to a singular line.

Other areas of the wing did not require adjustments because the surface grid did not propagate any problems into the field grid. Cross-sections of the grid can be seen in Figure (45) through Figure (56). The only other adjustment made was to stretch the entire grid to a suitable distance from the wing. This also was done with a small program that operated on the data file that was generated from the hyperbolic grid generator. While the final grid appears more uniform throughout the entire field, slight deviations in the orthogonality to the surface did occur. The final grid dimensions for the spherical field grid were 160x240x68 and can be seen in Figure (42) through Figure (56), Appendix C.

C. PARABOLIC GRID GENERATION

The last grid generation technique based on the solution of P.D.E.s is the parabolic grid generation method. The parabolic grid generation techniques may be constructed by modifying elliptic methods and hence carries various advantages of the method. The most popular modification is the elimination of the second derivatives. The solution is generated by marching out in one direction like in the hyperbolic method, but the marching is influenced somewhat by the other boundary. Control functions can be used to enhance orthogonality, which would not occur normally. Because of the effect of the other boundaries these methods tend to have more smoothing effects than a hyperbolic method. The parabolic method has the characteristics that are present in both elliptic and hyperbolic grid generation methods. The complexity tends to be less than the elliptical method and hence is faster.[Ref. 4 : pp. 277-278]

V. RESULTS AND DISCUSSIONS

Modern fighter aircraft designs take advantage of the strakes to improve controllability and enhance lift capabilities at high angles of attack. Existing aircraft are currently structurally modified by adding leading edge extensions to wings in order to improve the fuselage flow field characteristics which would eventually lead to improved lift and maneuverability characteristics [Ref. 28] . Complex fluid dynamic phenomena are associated with the high angle of attack flow over the forebody, the strake and the wings. The flow separates to form vortices which provide nonlinear lift. At high angles of attack the forebody, the strake and the wing vortices interact with each other and as a result self-excited unsteady flow may be triggered. When the angle of attack is further increased vortex breakdown occurs which will enhance flow unsteadiness and may result in a loss of controllability and other undesirable effects such as wing rock or tail buffet. Many of these interesting flow phenomena can be observed for the flow over the strake-delta wing configuration model for which the grid was generated.

In this chapter a survey of the grid generation will be done and the results of the numerical solution showing the characteristics of the flow field will be presented. Discussions on vortex formation, interaction and breakdown will be made for the various angles of attack.

A. GRID GENERATION

The generation of the surface and the field grid is a prerequisite for the numerical solution. The grid generation can be a very time consuming process. However, the grid quality will determine the accuracy of the numerical solution. The double-delta wing analyzed here is a simple model of a modern fighter aircraft planform. The surface definition can be done entirely using linear relationships. This allowed to use relatively simple algebraic and geometric relationships to generate a surface grid. Even with these simplifications the time expended on creating a surface grid and two types of field grids was quite long. The amount of effort that is expended on grid generation for an actual aircraft configuration would very likely be more than one year.

For symmetric bodies it is sufficient to generate half the surface grid. The generation of the surface grid was simplified by dividing the wing into similar sections and writing computer programs specific for the cross-section. It was found to be simpler to progress from the nose to the tail by a Δx increment and compute the points (in the

yz-plane) that represent the wing cross-section at that particular x-location. The major concern during this phase was to write a code that would permit a variable distribution of the grid lines. It was also equally important that the distribution of grid lines must transition as evenly as possible between fine distributions and coarser distributions. Fine distributions occur at locations where large numbers of grid points are required. Coarse distributions were chosen at locations where small grid density suffices to capture the physics of the flow. While a finished surface grid may appear to have smooth distributions and be very uniform, only the subsequent field grid generation revealed whether the chosen distributions were adequate. Therefore, great emphasis should be placed on surface grid versatility in the early stages.

The distribution of the grid lines and the number of grid lines in the x-direction was changed during the interactive process of improving the surface grid to produce a suitable field grid. In the case of the cylindrical grid generation, this trial and error process was relatively short because the field grid is composed of two-dimensional grids. On the other hand, for spherical grid generation both axial and circumferential surface grid distributions were much more sensitive because they must be suitable for the generation of a three-dimensional mesh. During the process of generating both the cylindrical and spherical type grids, various small programs were written to refine and improve a particular grid that was generated. Constructing a grid requires a knowledge and familiarity with grid generation codes and some prior knowledge of the flow characteristics. For example, the cylindrical grid required the deletion of the first points of the original surface grid in order to eliminate the singularity at the apex. Another program was then written to extend the grid upstream to where conditions of the freestream where expected. These programs can be found in Appendix F. The generation of the spherical grid required the writing of additional small programs in an attempt to algebraically modify regions of the field grid that had small discontinuities, see Appendix F. Diligence in changing the distributions and alteration of the nose region resulted in a smooth spherical field grid. It is believed that despite the larger amount of time spent for the generation of the spherical grid, a better quality grid compared with the cylindrical grid was obtained. However, because of a time constraint this could not be verified by obtaining a solution on the spherical grid. The cylindrical grid was used for the numerical implementation because a flow solution was desired for the presentation of this work.

A visual comparison of the two completed field grids reinforces the opinion that a spherical grid topology is more suitable for the subsequent numerical solution. It is

43

emphasized however, that spherical grid generation is very sensitive to the surface grid distributions and smoothness and requires larger computing times.

B. FLOW FIELD CHARACTERISTICS

The main characteristic of the flow over delta wings is the presence of the leading edge vortex. The nonlinear induced lift by the leading edge vortices has been actively investigated in recent years. Vortical flow is an advantageous lift generation mechanism that can be utilized successfully at medium to high speeds. A description of the leeward-side flow field characteristics will be presented and the flow field structure over a double-delta wing at various angles of attack will be analyzed. Available experimental results will be used to validate the computed results.

1. Vortex Characteristics

The leading edge vortices result from the roll-up of the shear layer that is shed from the leading edge. At moderate to high angles of attack the wingward and leeward side flow merges to form a shear layer that rolls up and forms a rotational vortex core. In the case of the double-delta wing a system of two primary vortices is formed. The first is along the strake and the second along the leading edge of the wing section. The primary strake vortex reattaches itself to the centerline of the planform. The vortex strength is increased by a continuous feeding of vorticity from the shear layers of the leading edge. Surface pressure suction peaks are produced at a location below the position of the vortex core. Because the vortex core exhibits large gradients of vorticity and circumferential velocity, large viscosity effects are expected. As the vortex strength increases downstream so do the lateral velocities that are near the surface. Coincidental with large velocities is a decrease in pressure. A secondary separation and formation of the secondary vortex is the result of these large lateral velocities and the associated adverse pressure gradients. If the secondary vortex is strong enough, a tertiary vortex can form under the secondary vortex by the same mechanisms [Ref. 17]. A schematic showing the leeward side vortex system and sense of rotation of these vortices is shown in Figure (6). Separated flow of the secondary vortex system reattaches again on the wing leeward surface. This separation and reattachment process can be best visualized with the use of surface oil flow patterns, see Figure (57). This visualization method has effectively shown the separation and reattachment of both primary, secondary and tertiary vortices.

The strength of the leading edge vortex increases downstream and as the angle of attack is increased. The pressure gradient in the direction of a vortex core accelerate

44
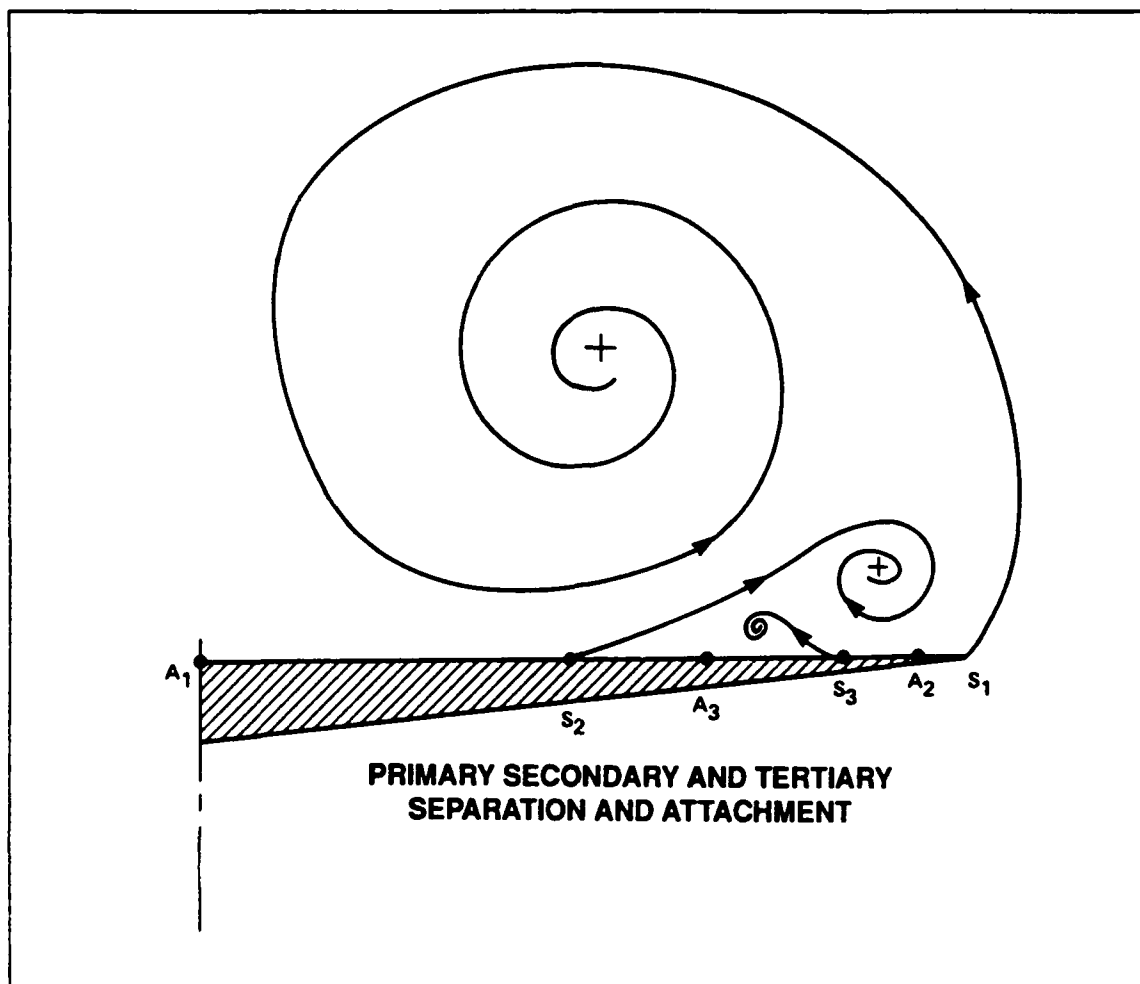
Figure 6.   Primary, Secondary and Tertiary Vortices

the fluid particles until a critical angle of attack is reached. At this angle the organized vortex core suddenly breaks down due to the adverse pressure gradient at the trailing edge. . This sudden transition is more commonly known as vortex burst or vortex breakdown. Vortex burst is a flow phenomenon that needs to be understood because a loss of the suction peak will occur and this change of the induced lift can result in undesirable effects on the aircraft. Work in this area by Sarpkaya, Thomas, Kjelgaard and Sellers, Ekaterinaris, Hawk, Barnett and O'Niel, Kegelman and Roos and many others has been conducted in recent years [Ref. 12,29,30,20,31,32].

45

Vortex burst is a transition from a jet-like spiraling flow to a wake-like flow. Adverse pressure gradient and swirl angle of the flow contribute to this phenomenon. Swirl angle in defined as;

$$\phi = \tan^{-1}(\frac{u}{v_\theta})$$

where u is the axial velocity component and v_θ is the azimuthial velocity component. Vortex breakdown will usually occur when the swirl angle exceeds a critical value of approximately $40°$. The swirl angle and the adverse pressure gradient determine the type of burst for a cylindrical vortex, i.e., bubble breakdown, spiral breakdown or double helix breakdown [Ref. 12]. Normally, as the angle of attack is increased the burst location will move upstream. If the angle of attack is increased even more, wake type of flow behind a bluff body will be encountered. Most of the initial investigations of vortex generation, induced lift and vortex breakdown was completed using a single-delta wing configuration. In this investigation the complex flow field that results due to the presence of multiple vortices is even further complicated for the case of the double-delta wing. This is due to the interaction of the strake vortex with the wing vortex and with the surface of the wing. The characteristics are shown for various angles of attack and results are discussed below in more detail.

2. Double-Delta Wing Flow Characteristics

a. Angle of Attack - 10°

The computed surface flow pattern at $\alpha = 10°$ shown in Figure (57) does not indicate tertiary separation on the strake, while secondary and tertiary separation are shown on the wing. The leeward side flow characteristics of the double-delta wing show at 10.0° angle of attack are shown in Figure (58) and Figure (59). Two primary vortices are formed, the first is formed by the sharp leading edge of the strake and the second by the leading edge of the wing. Both of these vortices are continually fed by vorticity as they progress downstream which increases their strength. The sense of rotation for the primary strake vortex and primary wing vortex is the same as can be seen in the velocity vector diagrams in Figure (60) through Figure (62). Also clearly shown in these figures, are the primary and secondary vortices having opposite swirling directions. Depending on the angle of attack, both primary vortices may swirl around each other, but here the vortices remain separated. On the other hand, the wing tip vortex and the wing vortex do eventually merge as can be seen in Figure (58).

46

At 10° angle of attack the primary strake vortex separates and reattaches at the centerline of the wing, see Figure (57). It can also be seen that a secondary separation occurs near the leading edge of the strake. This secondary vortex also reattaches itself, but it is not strong enough at 10° to generate a tertiary vortex. The wing vortex and reattachment of the primary and secondary vortex can also be seen in Figure (57).

b. Angle of Attack - 19°

At this angle of attack the primary strake vortex is much stronger. This can be deduced by the presence of a tertiary vortex as seen in Figure (63). This increase of vortex strength would produce an increase of the induced lift. The wing vortex that develops is continually fed by two sources. One source is the shear layer that is connected to the wing leading edge and the other is the shear layer associated with the primary strake vortex. This relinquishment of vorticity by the primary strake vortex causes its strength downstream of the kink to remain constant or even to reduce. These two vortices eventually merge close to the trailing edge of the wing, see Figure (64). The vortex burst defines the limit of vortex strength that can be maintained by the flow field. The burst appears to occur shortly after the primary strake and primary wing vortices merge, Figure (64) and Figure (65). It is at this angle of attack, i.e. just before vortex burst appears, where the maximum induced lift occurs. As the angle of attack is increased further the strake vortex continues to get stronger but the burst point moves further upstream. Close examination of particle traces, see Figure (64) and Figure (65), actually shows the development of a wing-tip vortex. This wing-tip will eventually merge with the primary wing vortex. The flow direction for both the strake and wing primary vortices is the same, see Figure (66) through Figure (68). These cross-sectional views of the velocity vectors also show the counter rotation between primary, secondary and tertiary vortices. At the cross-section over the wing, Figure (66), the two vortex cores are distinct. While at the trailing edge, Figure (68), the two vortices are merged.

c. Angle of Attack - 22.4°

As the angle of attack continues to increase the most prominent feature is probably the change in location of the vortex breakdown. The location of the burst will move upstream as the angle of attack is increased. Not only does the burst location move upstream, but the strength of the strake vortex increases. Figure (69) clearly shows the development of a tertiary vortex which is characteristic for a strong vortex. The breakdown of the vortex is readily apparent in the 22.4° flow solutions. In Figure (70) and Figure (71) the bursting of the strake vortex over the wing can be seen. The

rotations of the vortices that develop here are the same as at smaller angles of attack and can be verified by Figure (72) through Figure (74).

3. Comparison with Experimental Data

Although the purpose of Cunningham and Boer's experiment was the investigation of unsteady phenomena, the report also presents steady state data. The development and location of the vortices are in qualitative agreement with Cunningham and Boer's flow visualization results [Ref. 18]. These authors also present steady pressure measurements for five different angles of attack. Unfortunately, insufficient time was available toward the completion of this investigation to attempt a detailed comparison of the present computational results with the pressure data. The only comparison made was for the pressures calculated at $\alpha = 19°$. Similarly, the double-delta wing studied by Krause and Liu [Ref. 17] and the experimental data of Brennenstuhl cited therein has not yet been used for comparison purposes.

A comparison of the pressure coefficient and spanwise location to experimental data can be seen in Figure (75) through Figure (77). Three axial locations on the wing were selected, specifically, $x/c = 0.40, 0.66, 0.98$. The locations were selected to investigate representative sections of the strake, the wing and the trailing edge. It can clearly be seen that at $x/c = 0.40$ there develop two suction peaks due to the primary and secondary vortex. The location of these peaks differs from the experiment, but the trend is well represented. Since the calculations are believed to be not yet fully converged, it is expected that more computational time will yield even more accurate results. It must be pointed out that these calculations are for vortical separated flow and the accurate capture of these characteristics is very difficult. In Figure (76), a cross-section of the wing shows the two suction peaks due to the primary strake and primary wing vortices quite well. As before, the calculations reproduce the trends quite well, but it is expected that more fully converged results will produce still better agreement. The last comparison is made at the trailing edge and is shown in Figure (77). Again, the trend is reproduced well: but, because at this axial location vortex breakdown occurs, it is very difficult to achieve exact results. This graph shows a smaller pressure coefficient compared to the other axial locations. This decrease is a direct result of the vortex breakdown.

All the comparisons of the coefficient of pressure depicted quite well the trends associated with separated vortical flow. It is expected that the locations of the suction peaks will be even more accurate after more convergence. The comparison made here are based on results that required 35-40 hours of CPU time on the Cray-YMP computer.

It is expected that another ten hours of CPU time would produce even more accurate results. The expected total CPU time for each angle of attack is approximately 50 hours. Due to insufficient time, the fully converged results could not be presented in this paper.

VI. CONCLUSIONS AND RECOMMENDATIONS

An investigation of the flow characteristics that are created in the fluid domain surrounding a double-delta wing at various angles of attack by a numerical approach was presented. The numerical solution of the Navier-Stokes equations requires the discretization of the flow domain by a smooth computational grid. Accurate representation of the flow physics by the grid points will directly affect the quality of the flow field solution. Therefore, the surface and field grid density and topology must be carefully chosen. With the flow field domain defined the numerical solution (in this case finite difference) can be implemented to investigate the flow characteristics or compare with experimental results.

An algebraic method of grid generation was selected for the defining of the surface grid and the source code is presented in Appendix E. The important precaution is that when developing the source code attempt to allow the flexibility of as many parameters as possible. The grid line distribution and number of grid points in a specified direction were found to be most important. Subsequent generation of the field grid will require the moving of grid lines to a distribution that will produce a smooth and continuous grid.

Different types of numerical grid generation techniques are available, such as hyperbolic, elliptic and parabolic. The advantages, disadvantages and characteristics of these methods were discussed. The hyperbolic grid generation technique was chosen and two field grid topologies were generated, a cylindrical grid and a spherical grid. The cylindrical grid was easier to generate, but the spherical grid yielded a smoother grid distribution in space. This was achieved at the expense of time and computational effort. However, use of the cylindrical grid would allow the generation of an acceptable flow field solution.

Once the grid was completed, a finite difference algorithm in conjunction with an algebraic turbulence model was utilized to obtain flow field solutions at $\alpha = 10.0°, 19.0°, 22.4°$ angles of attack. Investigations of vortex generation, vortex interaction and vortex breakdown were conducted. At moderate angles of attack the double-delta wing configuration showed primary vortices generated from both the strake and wing. These vortices produce nonlinear vortical lift which can be very beneficial to fighter type aircraft that operate at high angles of attack. At approximately 19° angle

of attack vortex bursting occurred just after the primary strake vortex and primary wing vortex merged together. The formation of the primary and secondary vortices over the double-delta wing compared favorably with the flow visualization data of Cunningham and Boer [Ref. 18]. It is strongly recommended, as the next phase of this investigation, to compare the present numerical results with the steady pressure data obtained by these two authors. Using the source code presented here as a building block, future studies could repeat the calculations for the double-delta wing studied by Krause and Liu [Ref. 17] and then compare with the experimental results of Brennenstuhl cited in Reference 17.

Future studies of this phenomenon might be to continue this same analysis utilizing the spherical grid to determine if the results are more accurate or if computational time is less, i.e. the solution field converges faster. Furthermore with the recent increased interest in dynamic stall phenomenon, an analysis could be done to compare the computational results of a pitching straked-wing to the experimental studies done by Cunningham and Boer. Because a major portion of time is spent in the generation of a field grid for an analysis of this type, the work load would be reduced by using the grid presented in this thesis.

The work load involved in generating a field grid is significantly increased when the body under investigation is an entire aircraft. For this reason the need for a method of quickly producing a surface grid would expedite a numerically generated solution of a flow field and allow more time for the improvement of numerical methods.
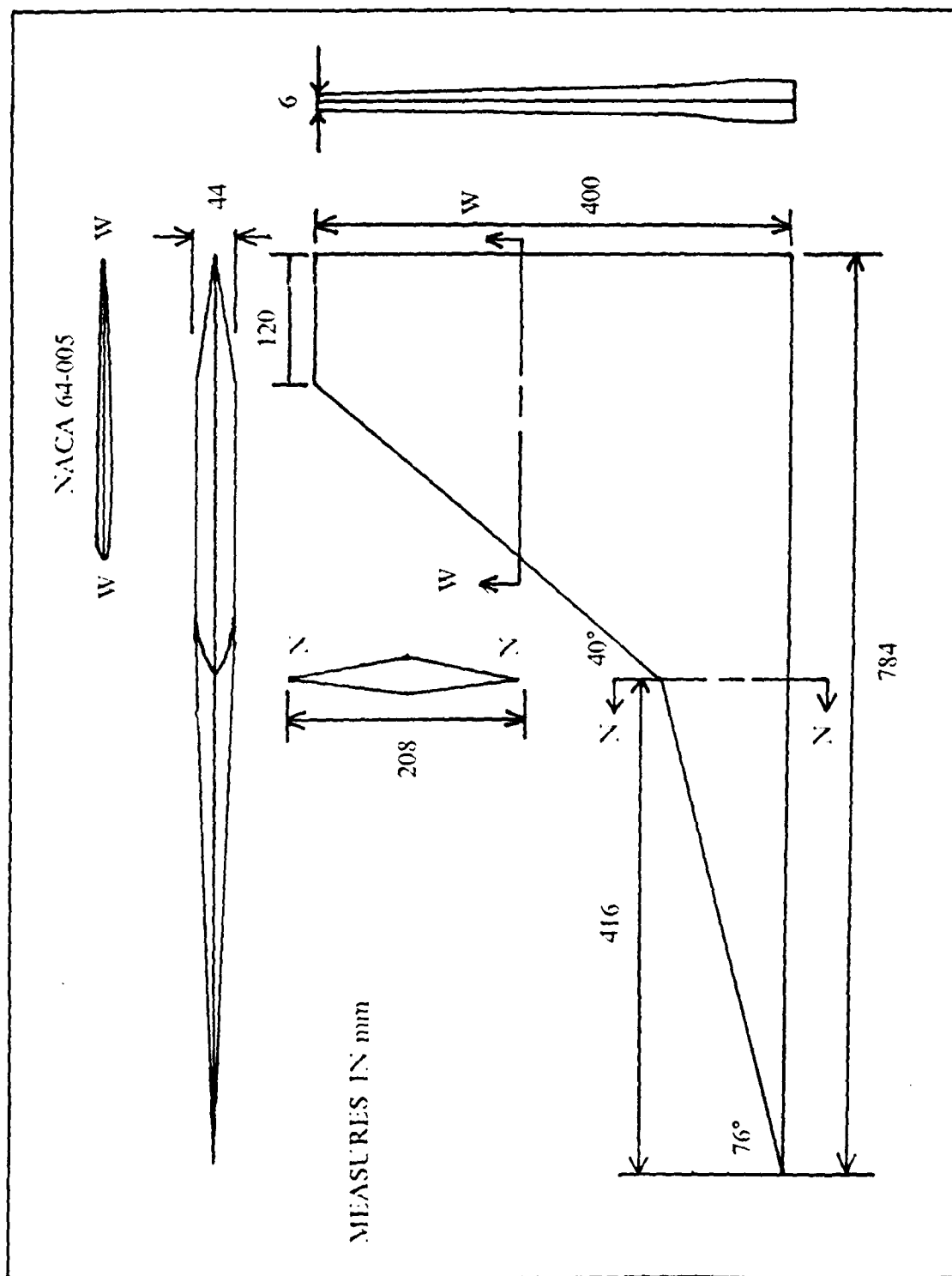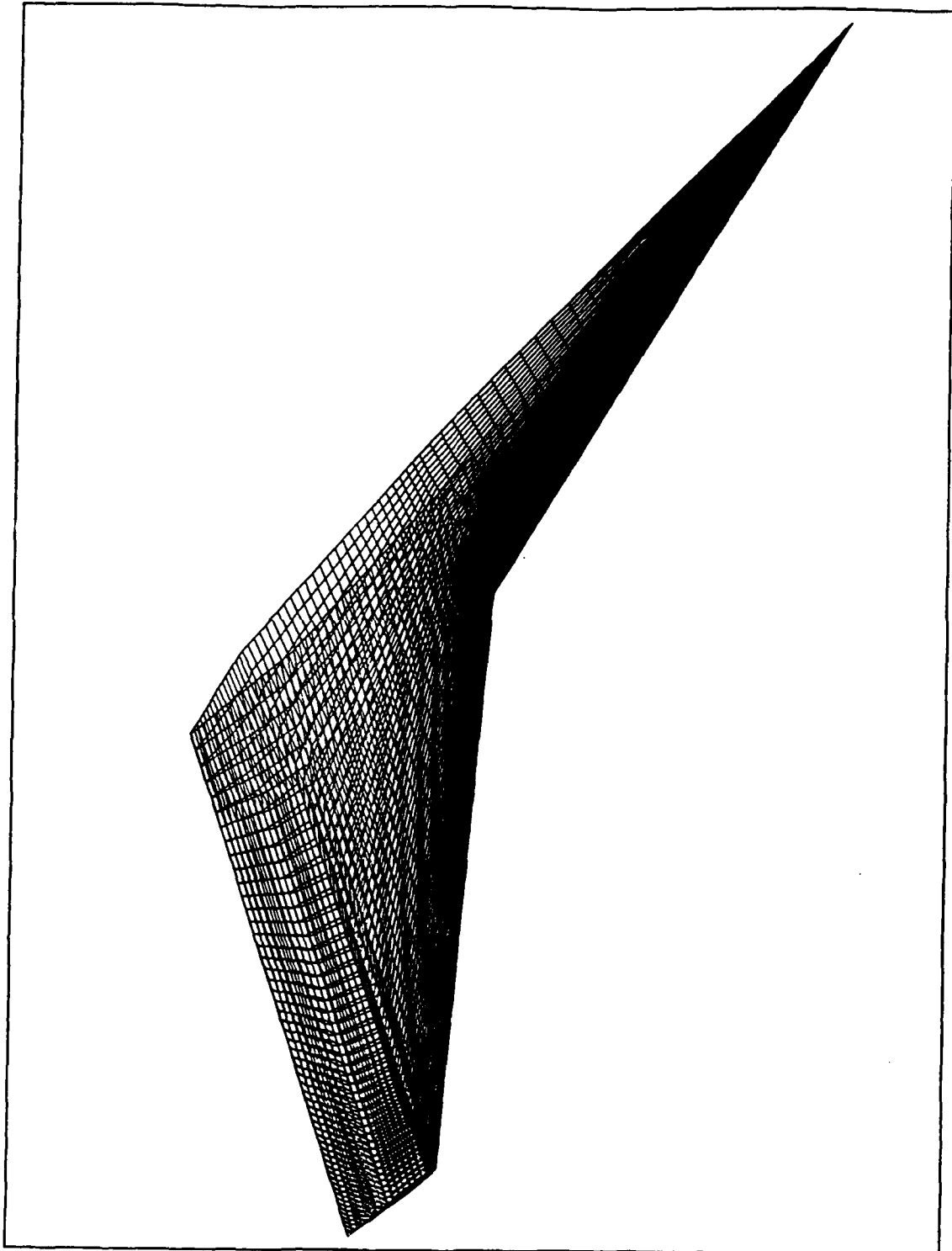
APPENDIX A.   SURFACE GRID FIGURES

Figure 7. Double-Delta Wing Configuration

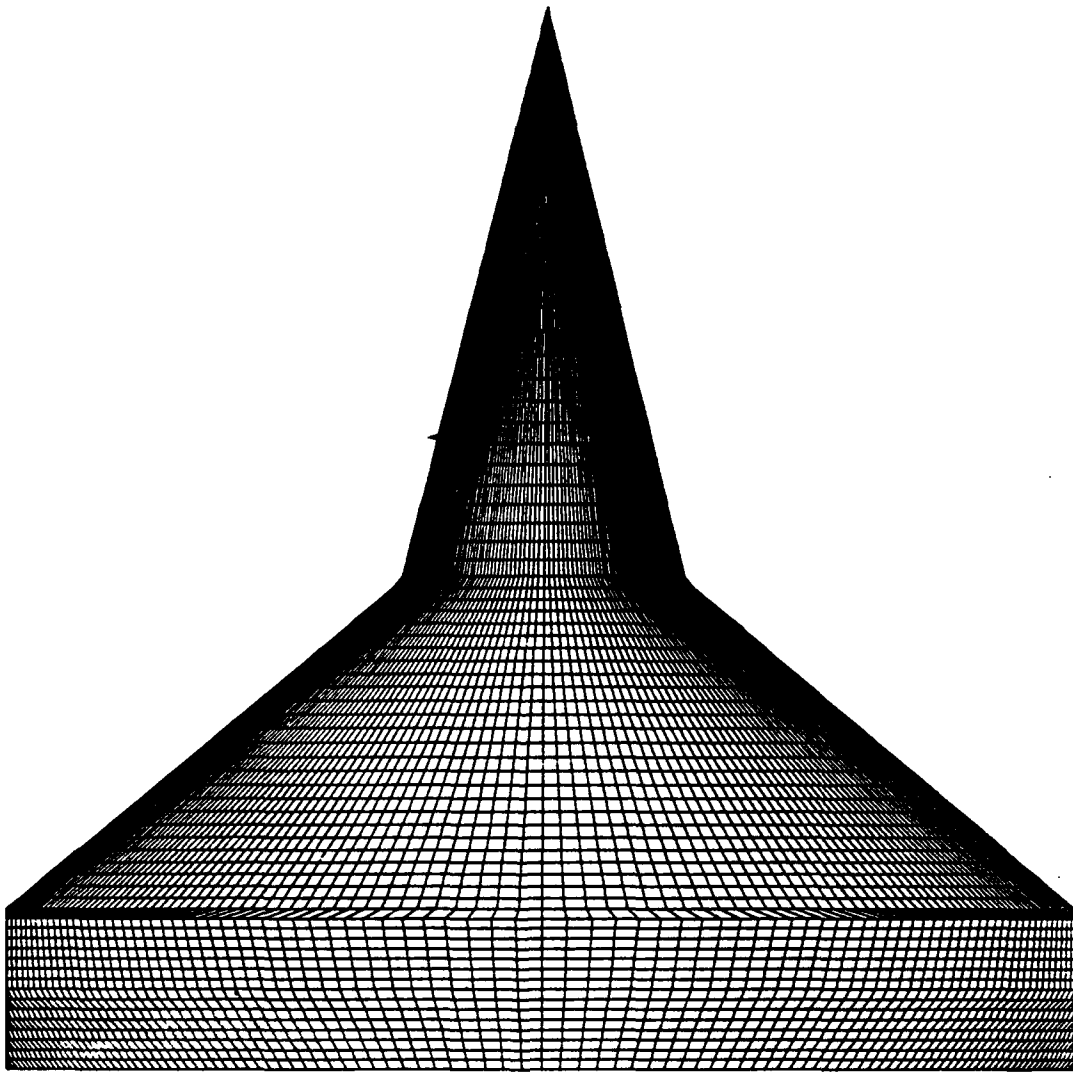Figure 8. Double-Delta Wing Surface Grid (120x240x1)
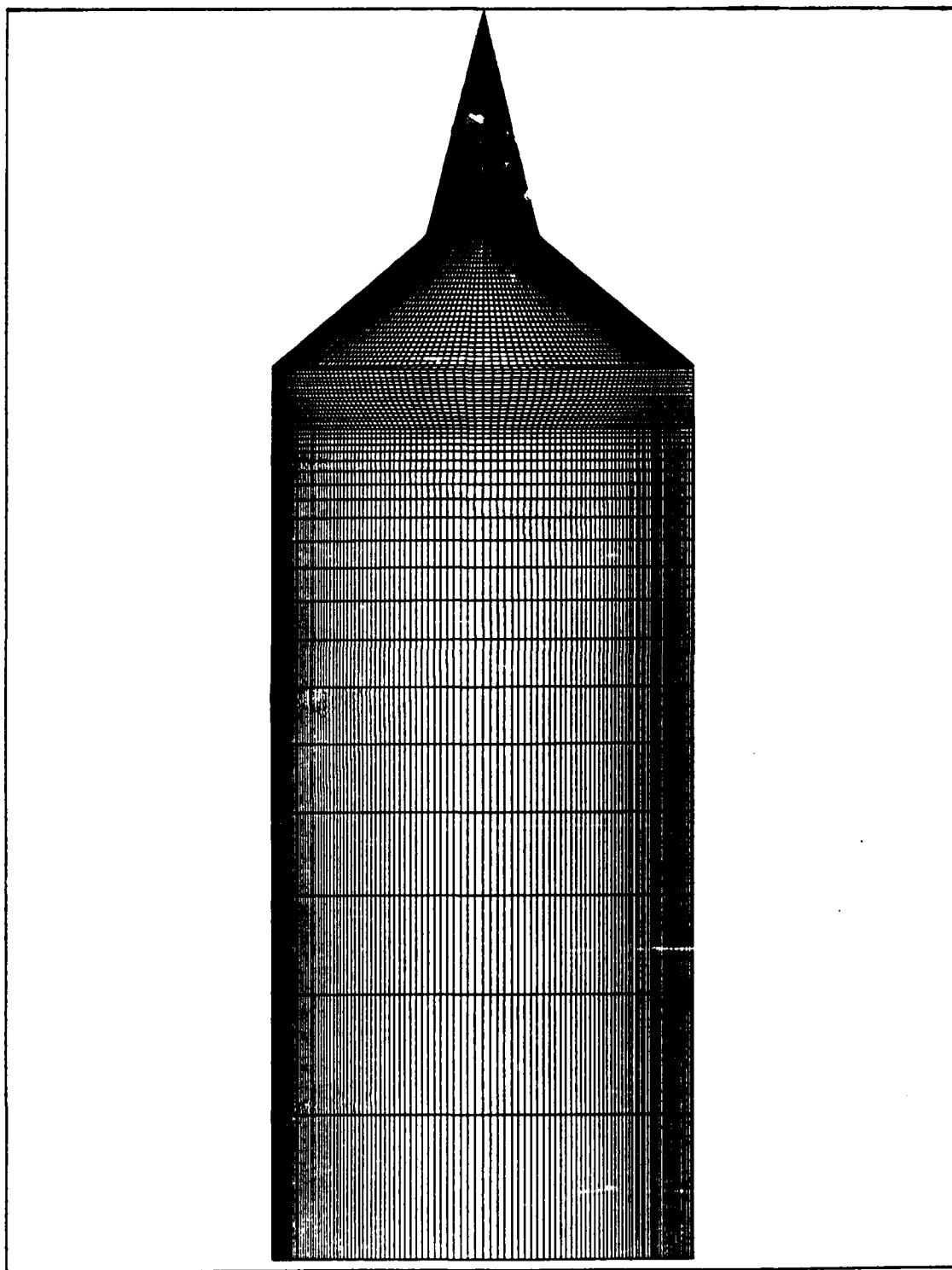
Figure 9. Double-Delta Wing - top view (100x240x1)

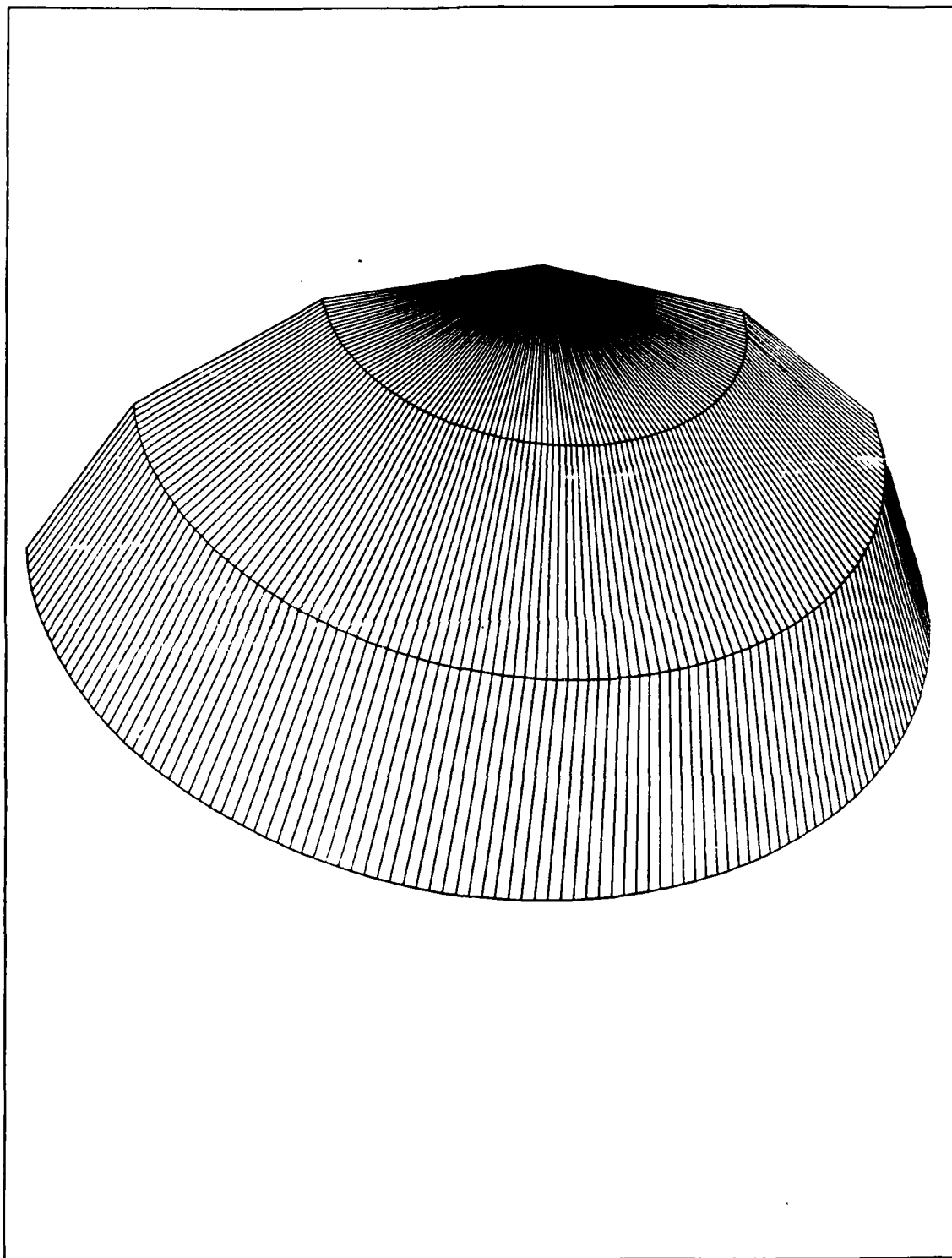Figure 10.   Double-Delta Wing and Wake - top view (120x240x1)
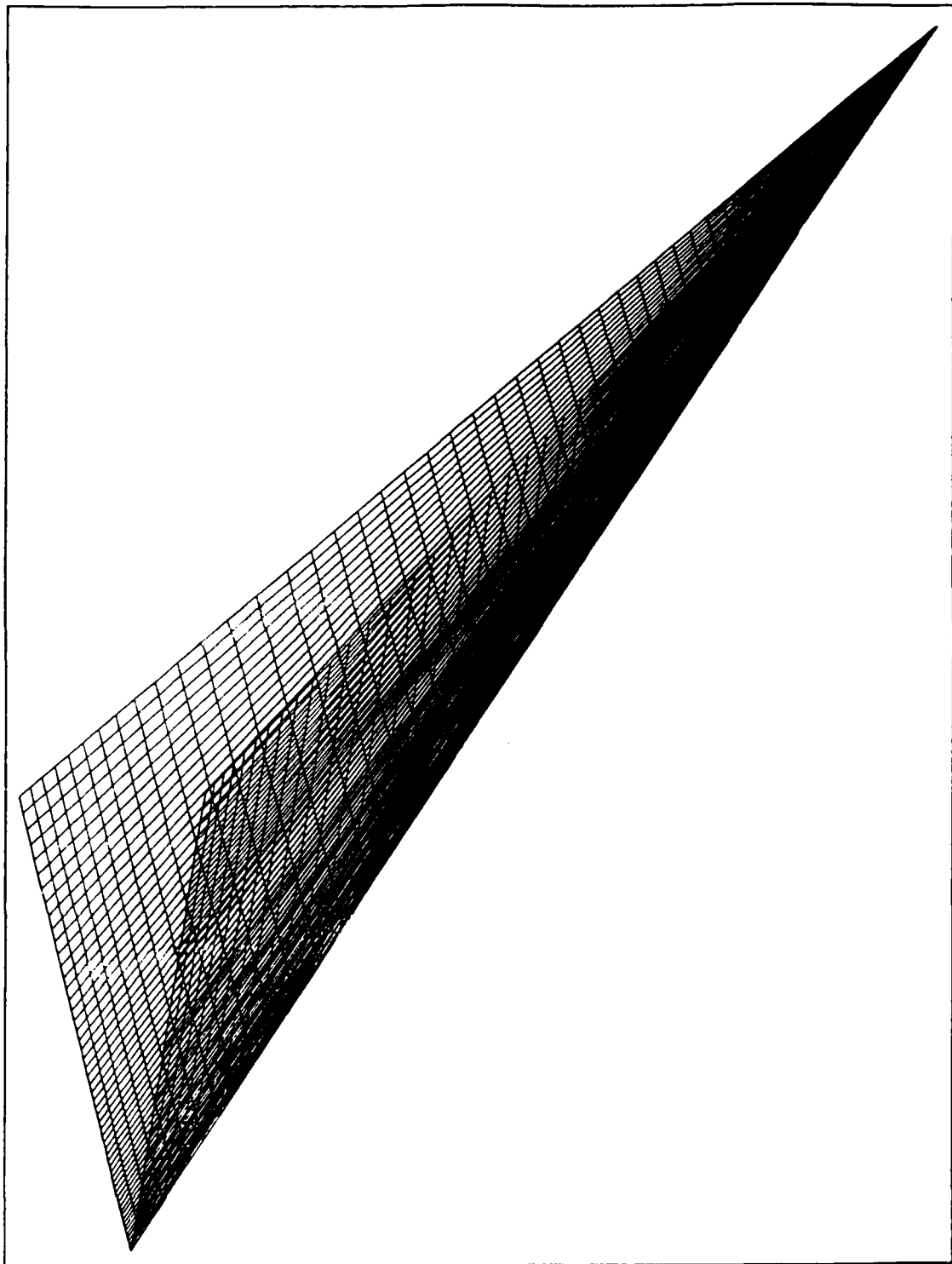
Figure 11.   Detail of Apex

57

Figure 12.   Grid Distribution of the Strake (48x240x1)
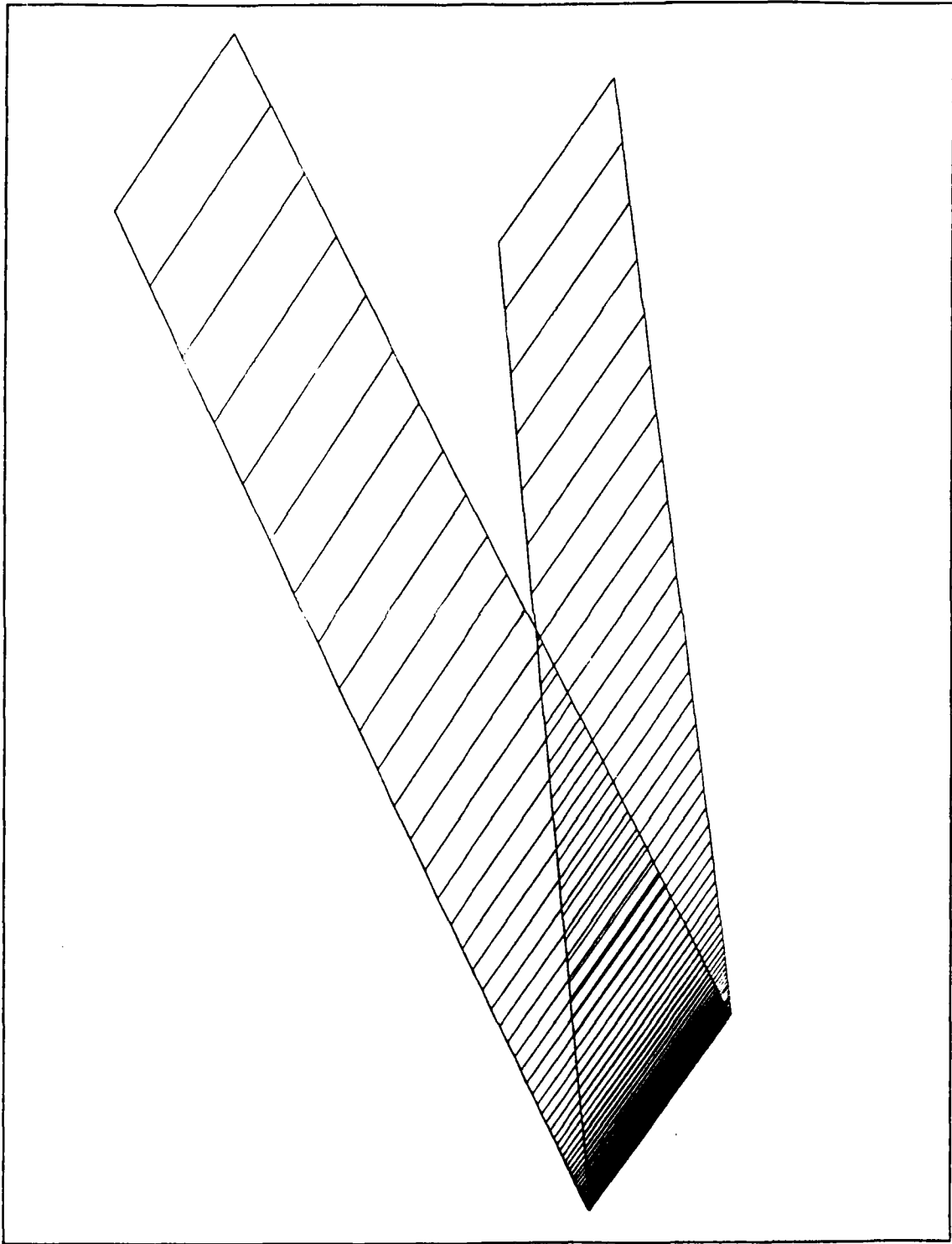
Figure 13. Typical Section of the Strake

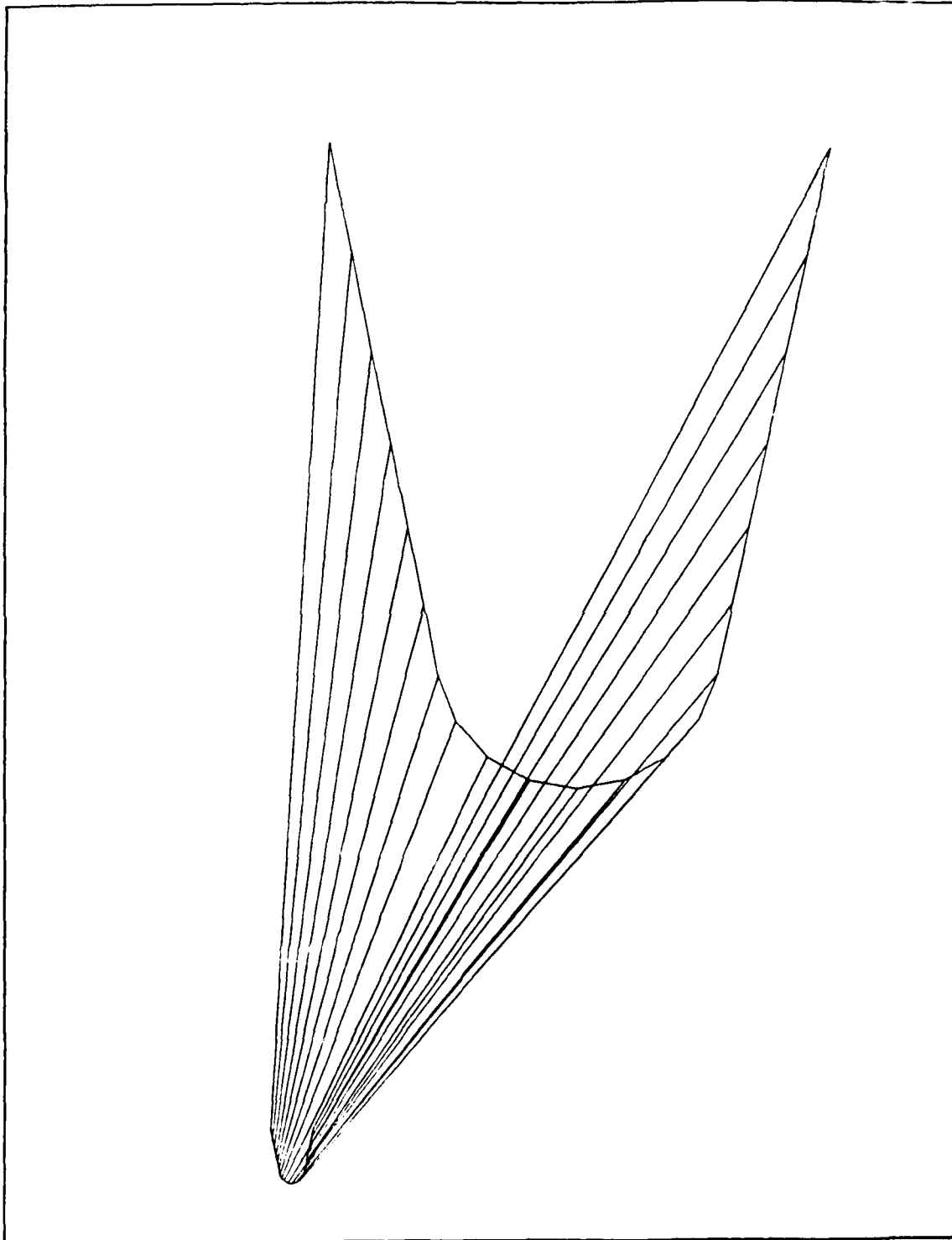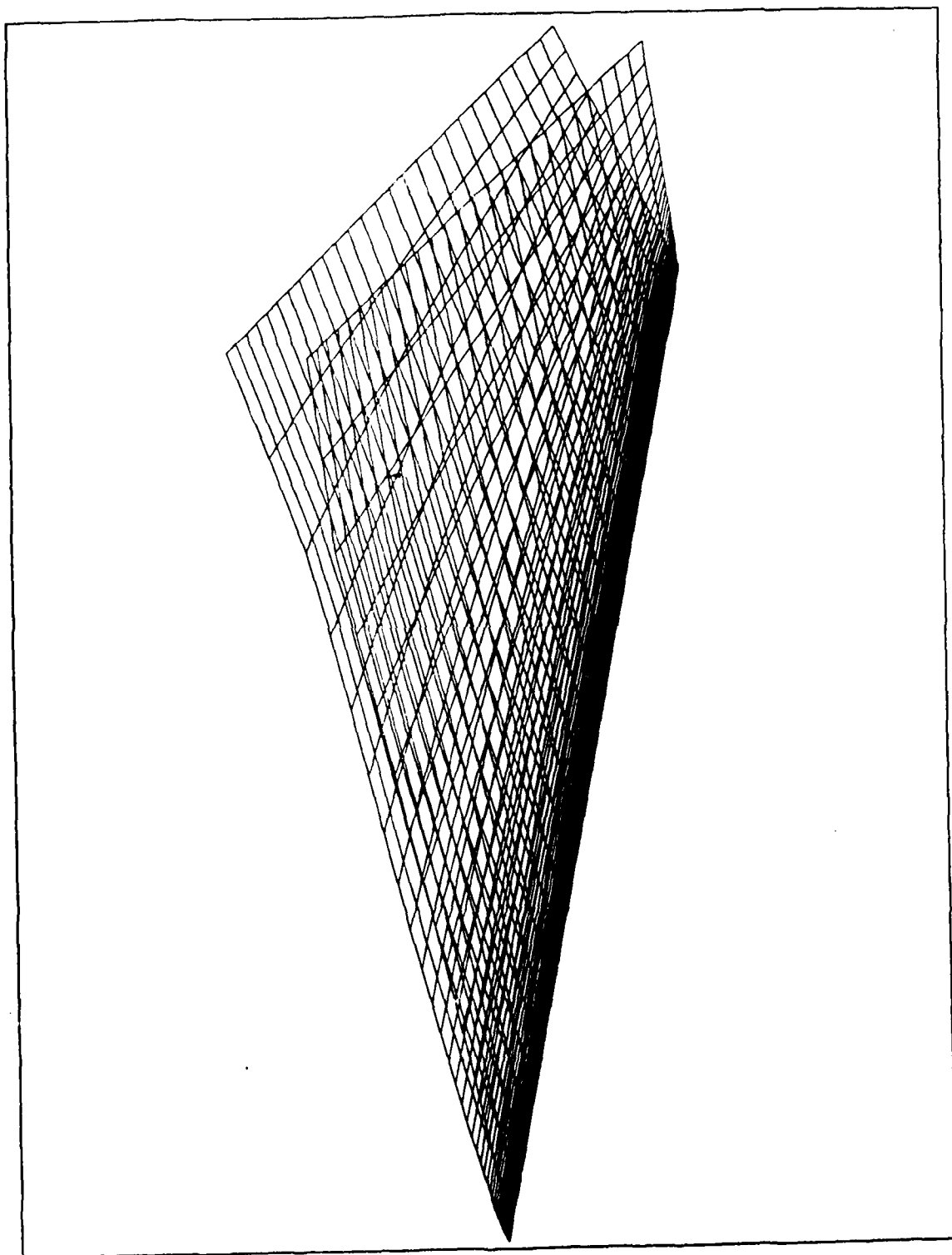Figure 14. Leading Edge Rounding of the Strake

Figure 15. Grid Distribution of the Wing (26x240x1)

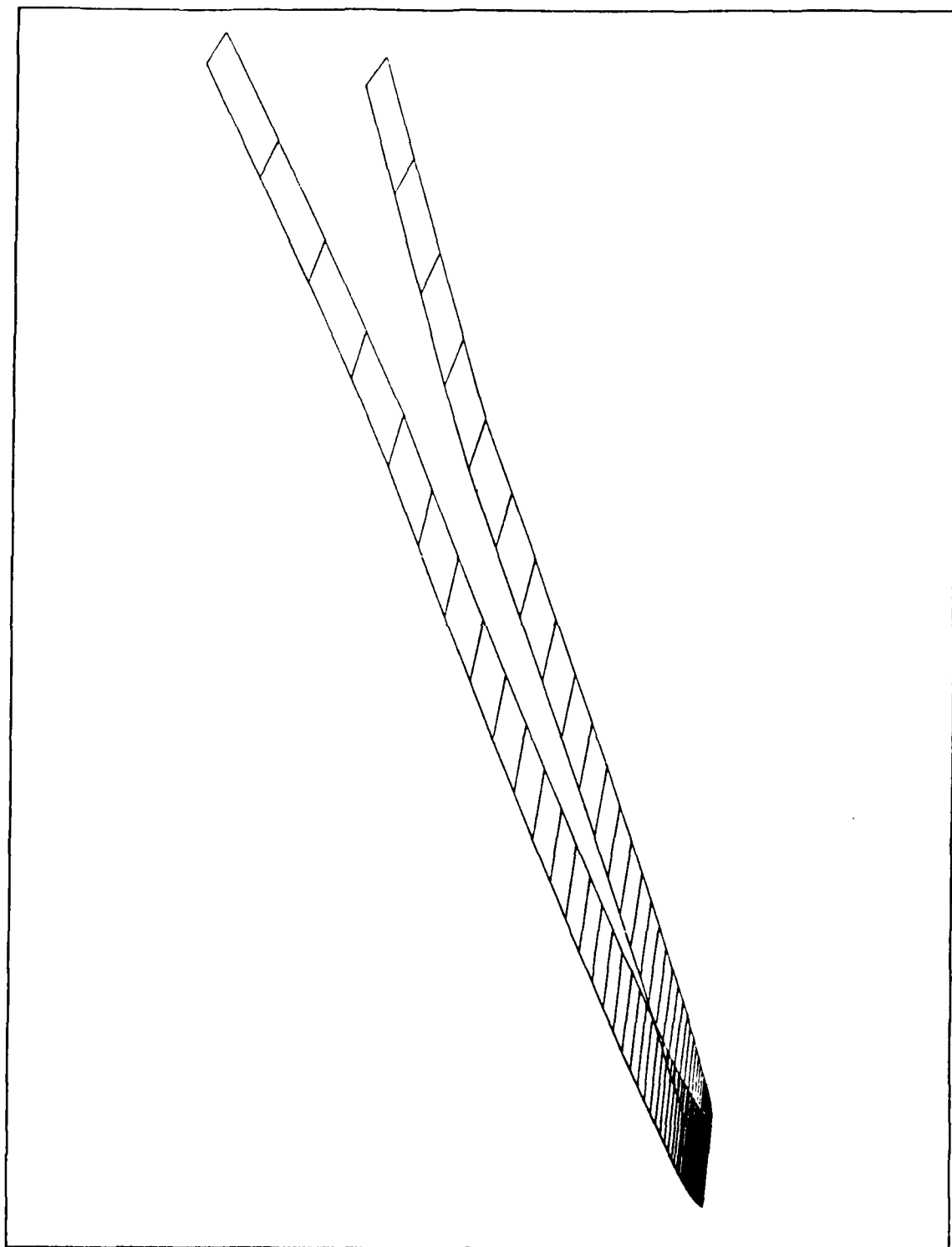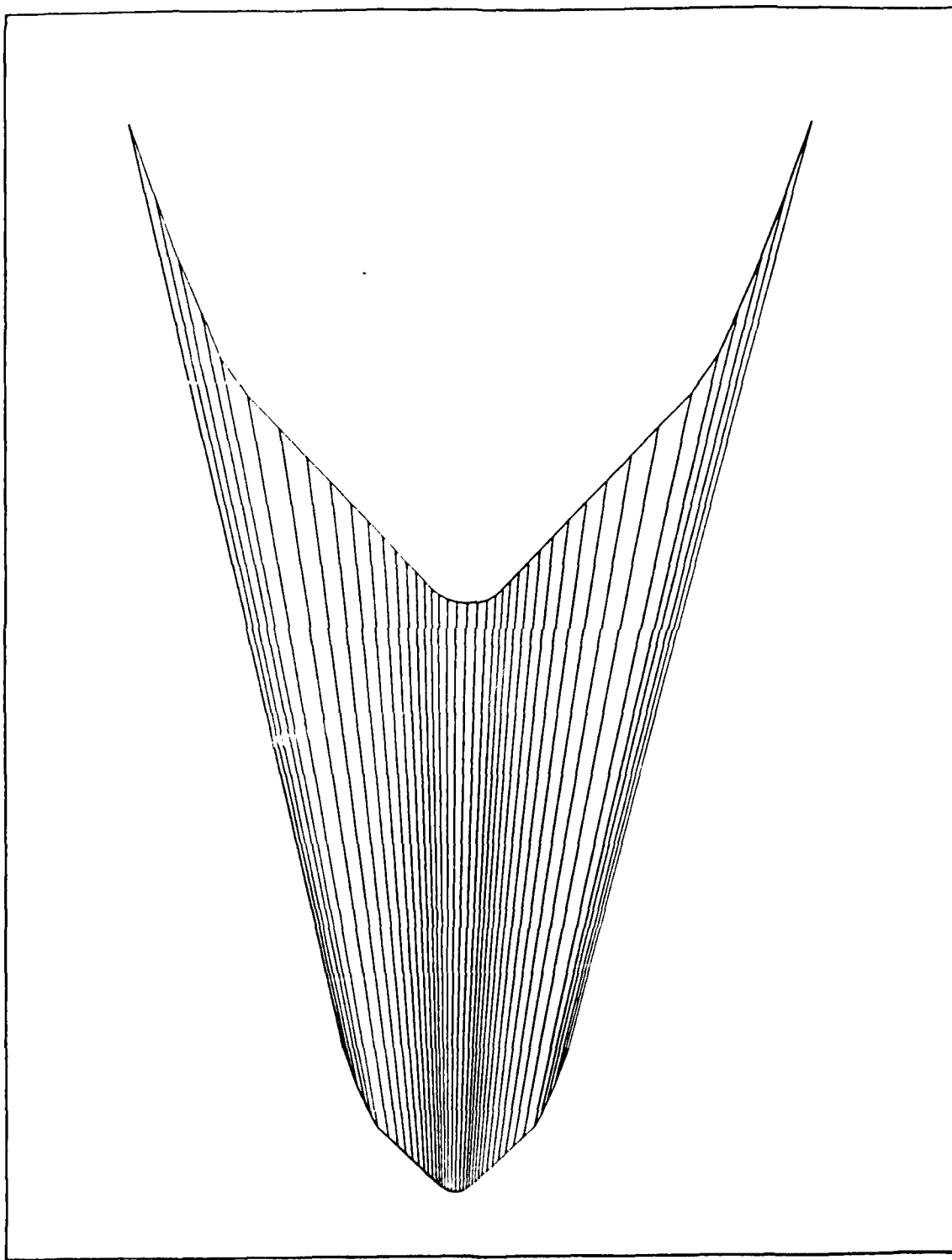Figure 16.   Typical Section of the Wing

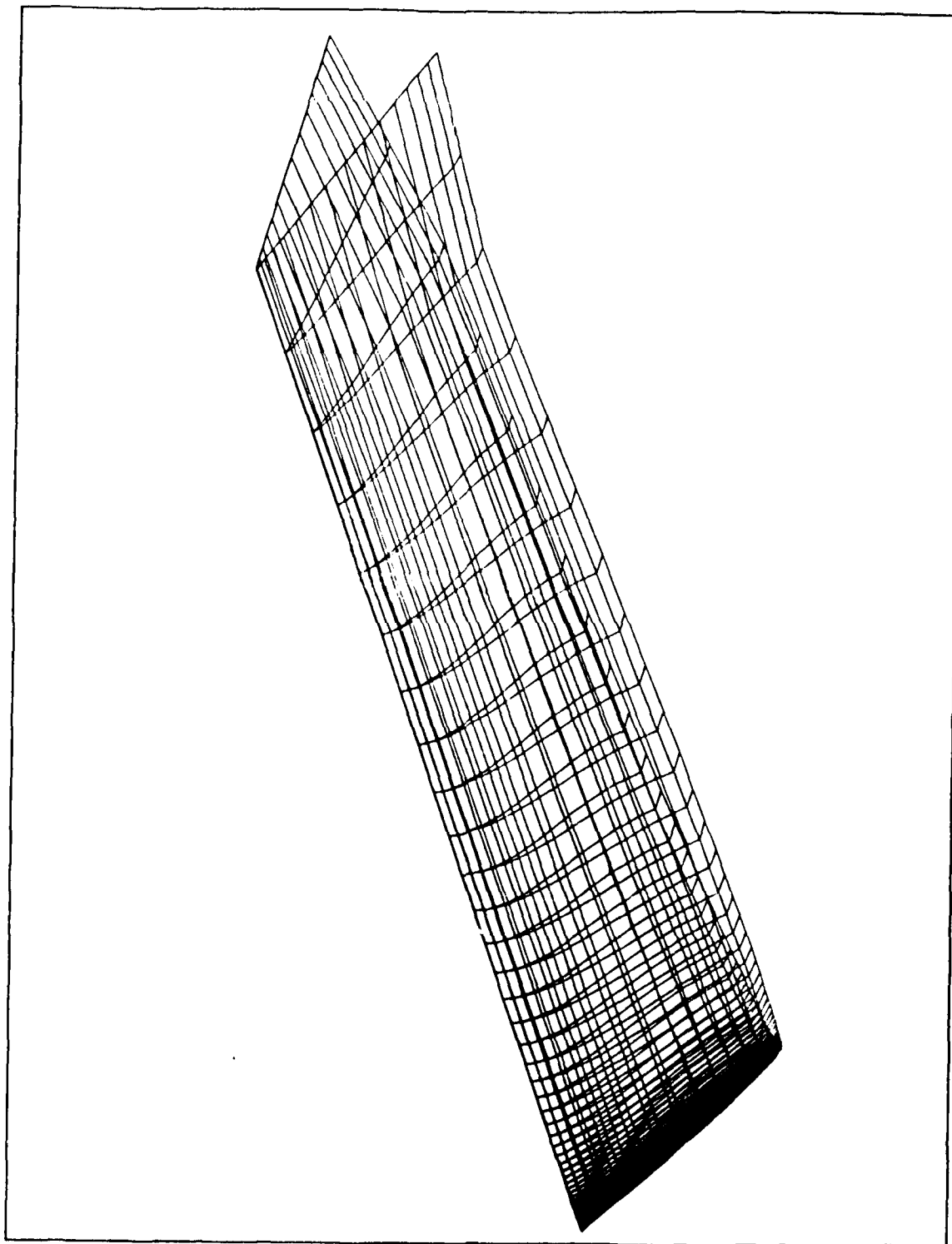Figure 17. Leading Edge Rounding of the Wing

63

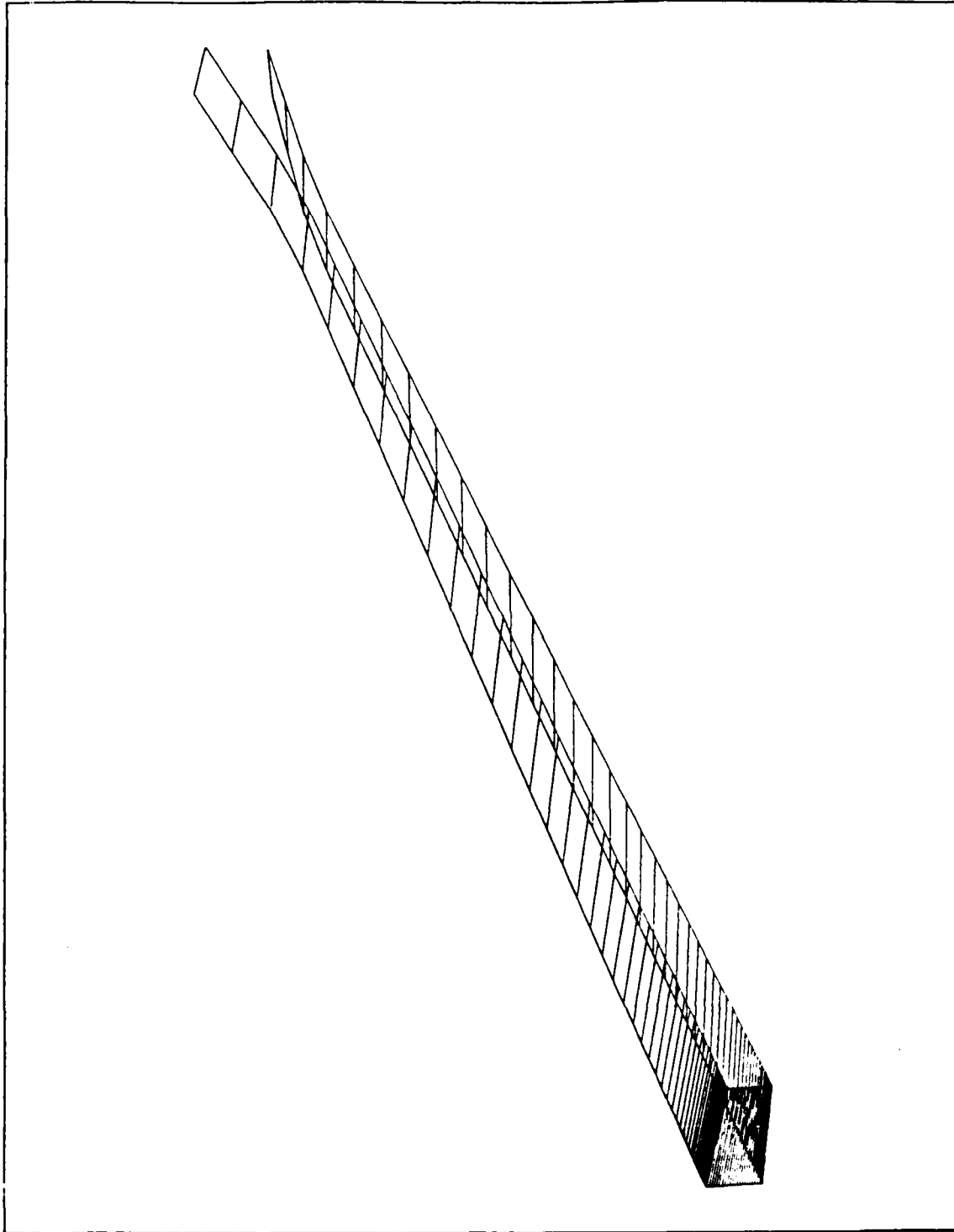Figure 18.   Grid Distribution of the Rectangular Section (16x240x1)

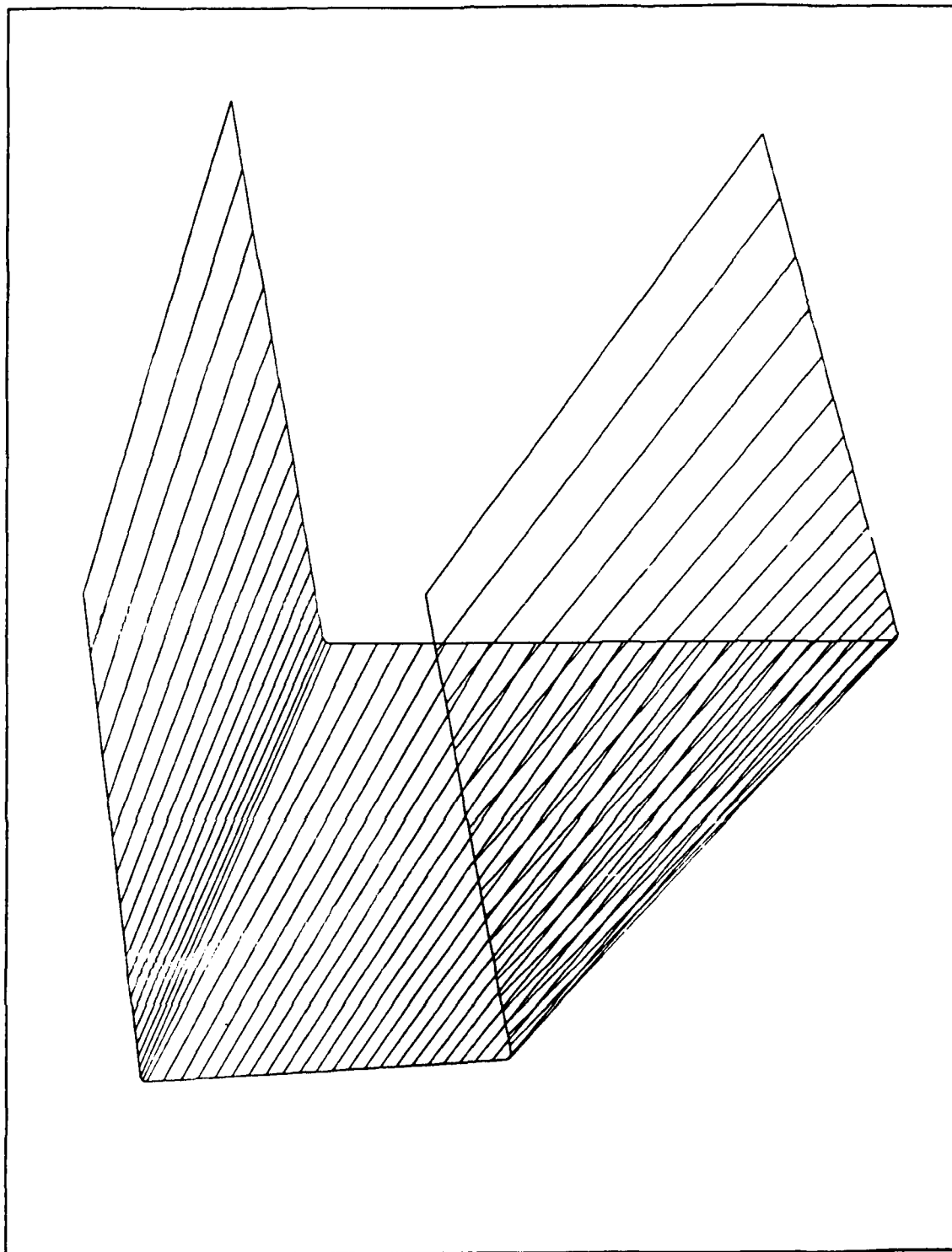Figure 19. Typical Section of the Rectangular Section

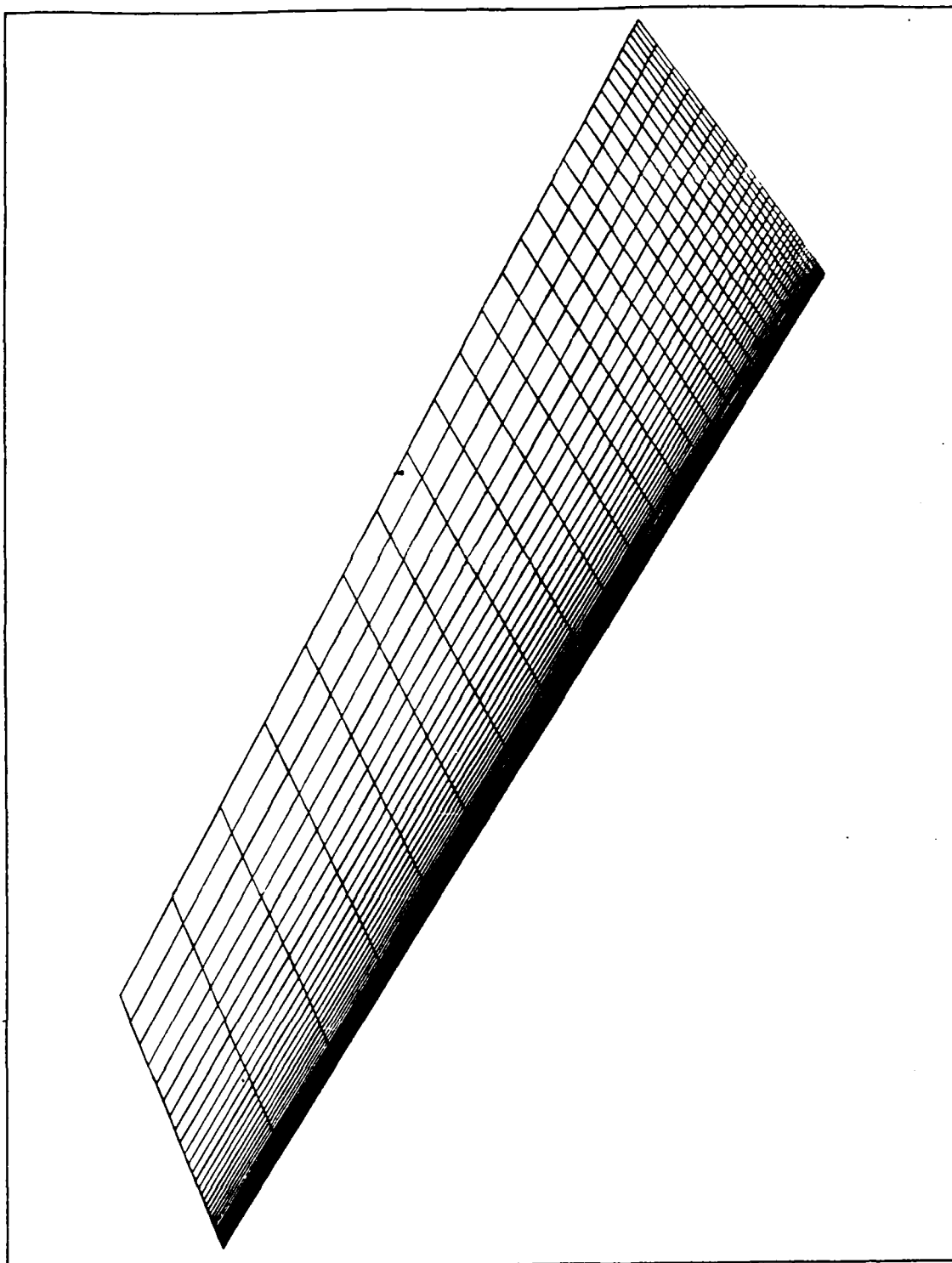Figure 20. Leading Edge Rounding of the Rectangular Section

Figure 21. Grid Distribution of the Wake (29x121x1)
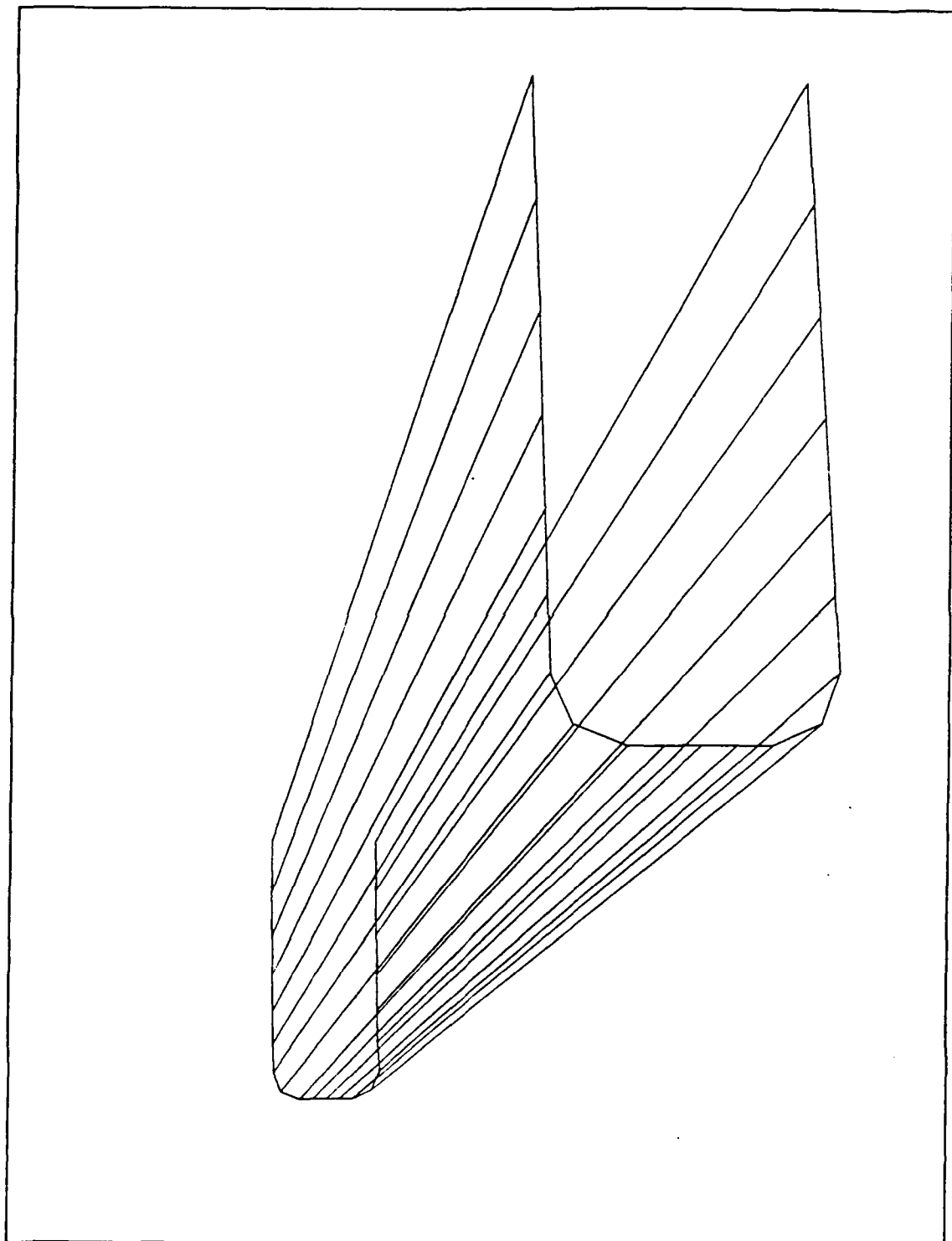
Figure 22. Edge Rounding of the Wake
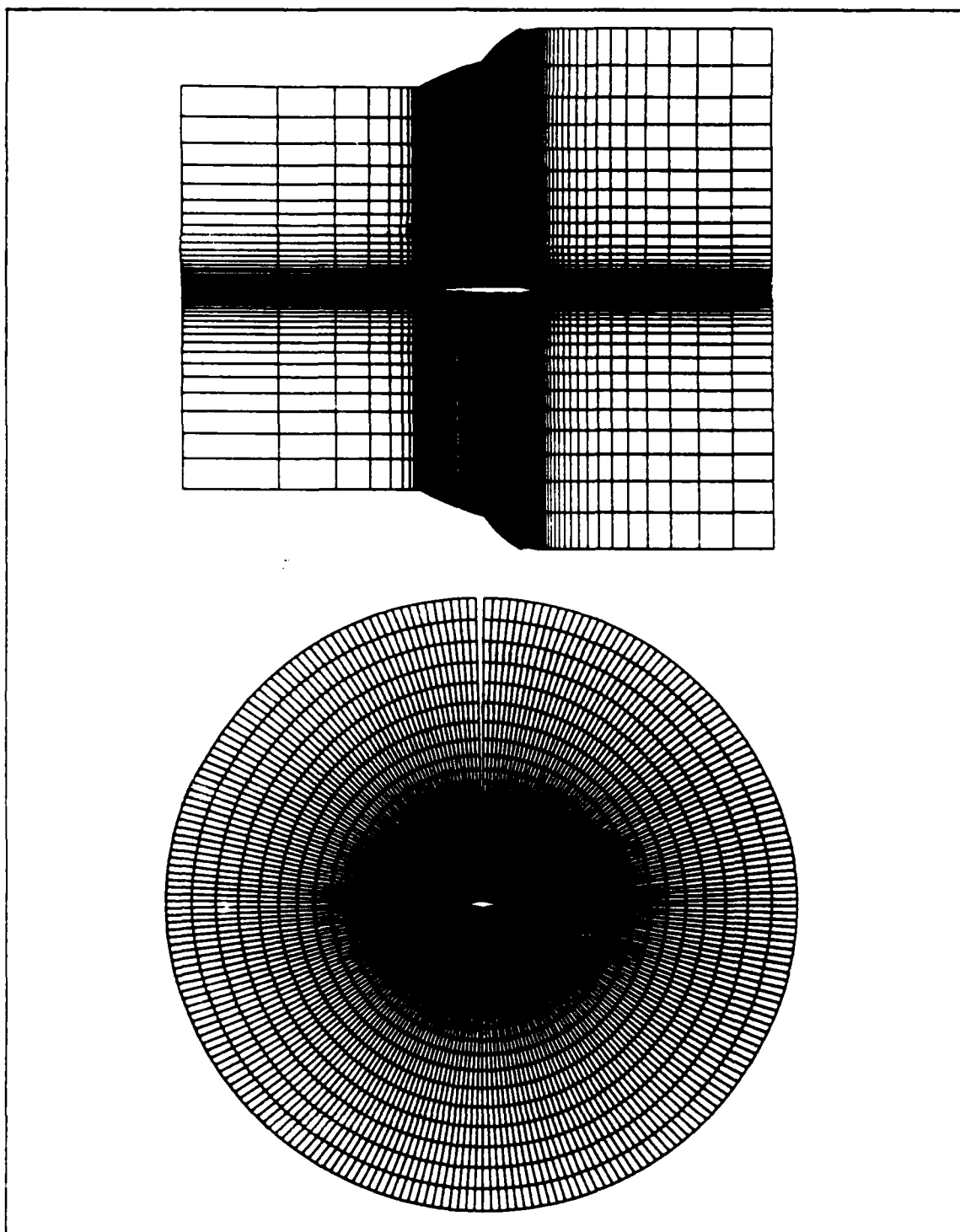
Figure 23.   Cylindrical (H-O) Grid Topology (130x240x68)
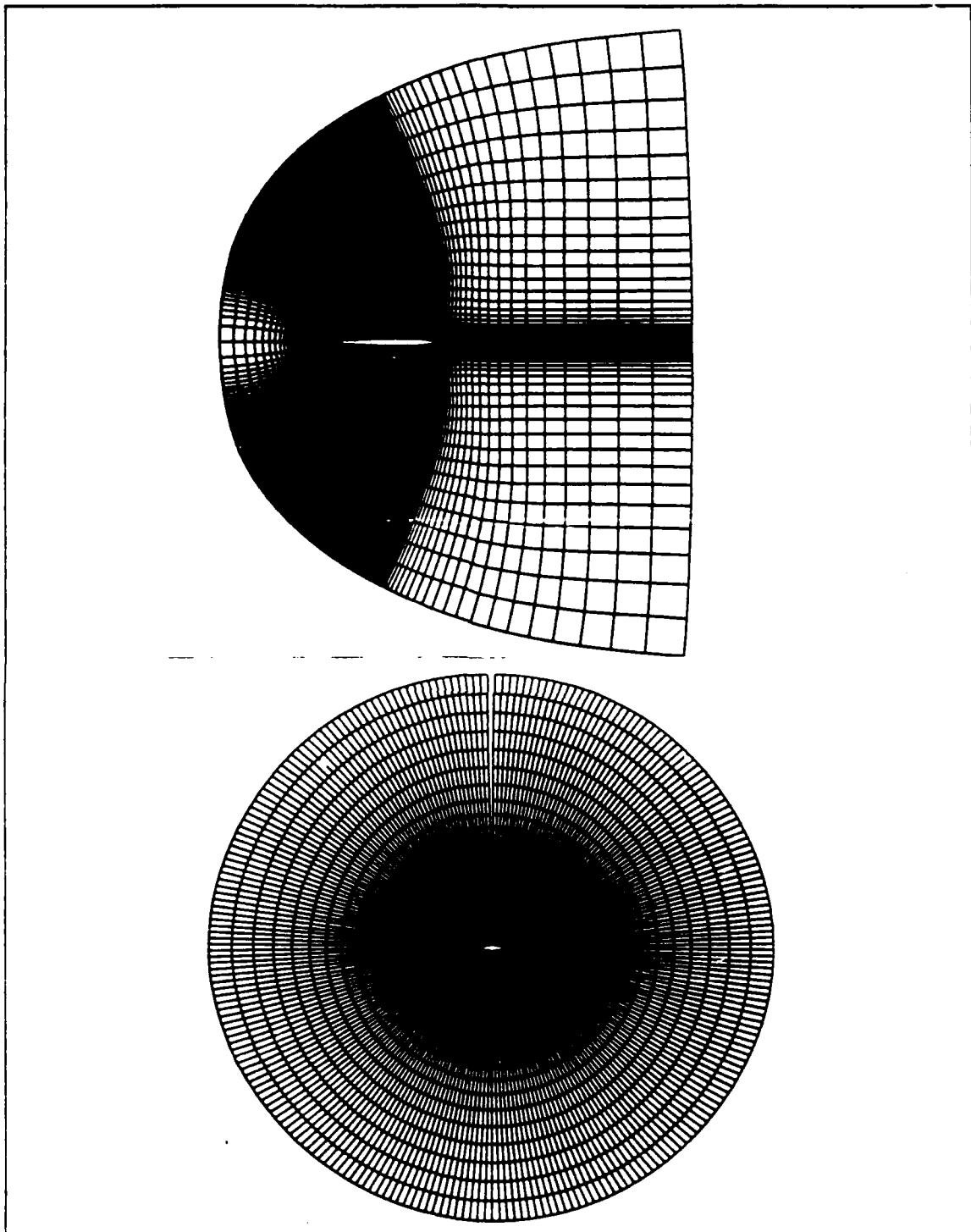
Figure 24.   Spherical (C-O) Grid Topology (160x240x68)

70

APPENDIX B.   FIELD GRID FIGURES -- CYLINDRICAL
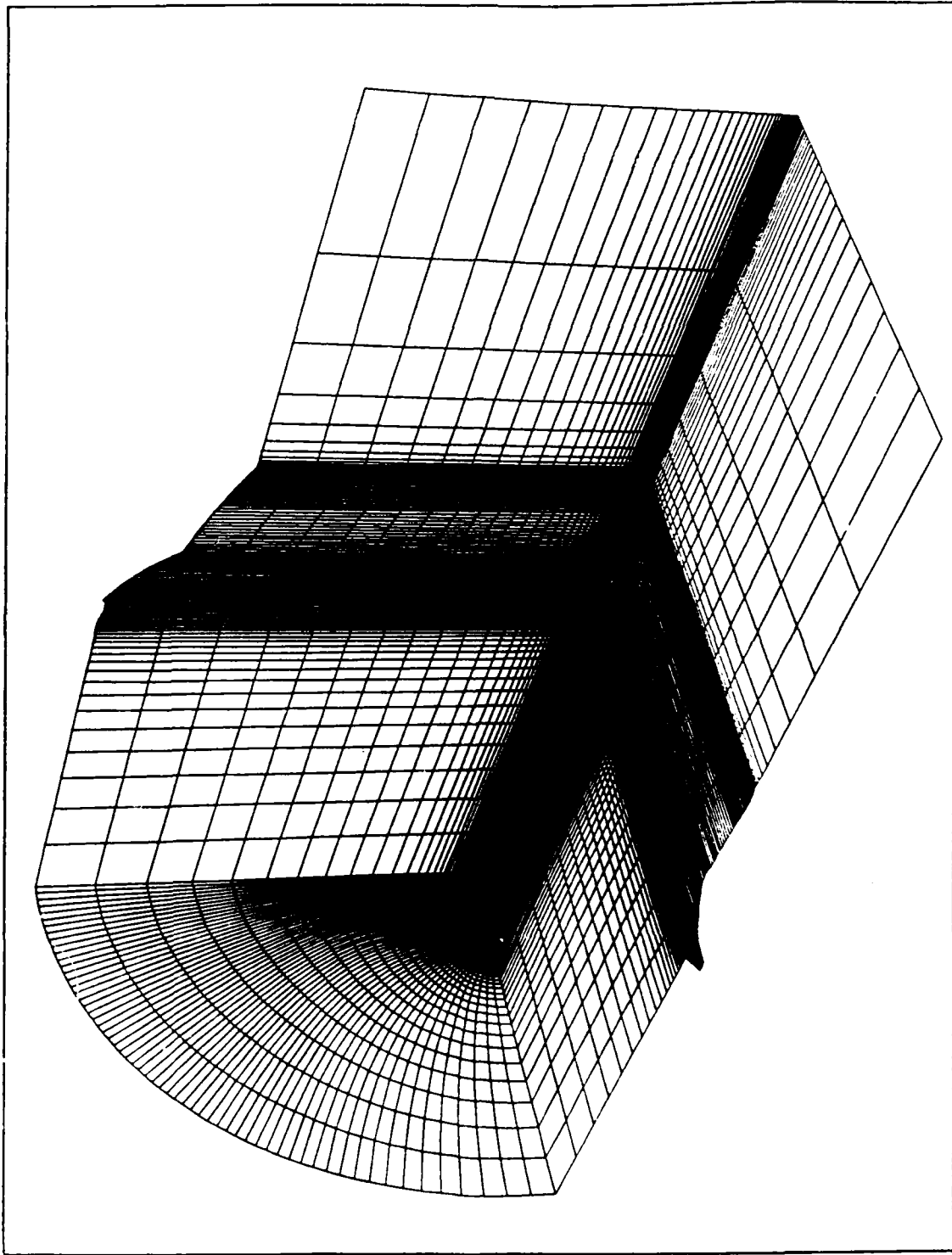
Figure 25. Cylindrical Grid Configuration (130x240x68)

72

Figure 26. Cylindrical Grid Configuration Detail (130x240x68)
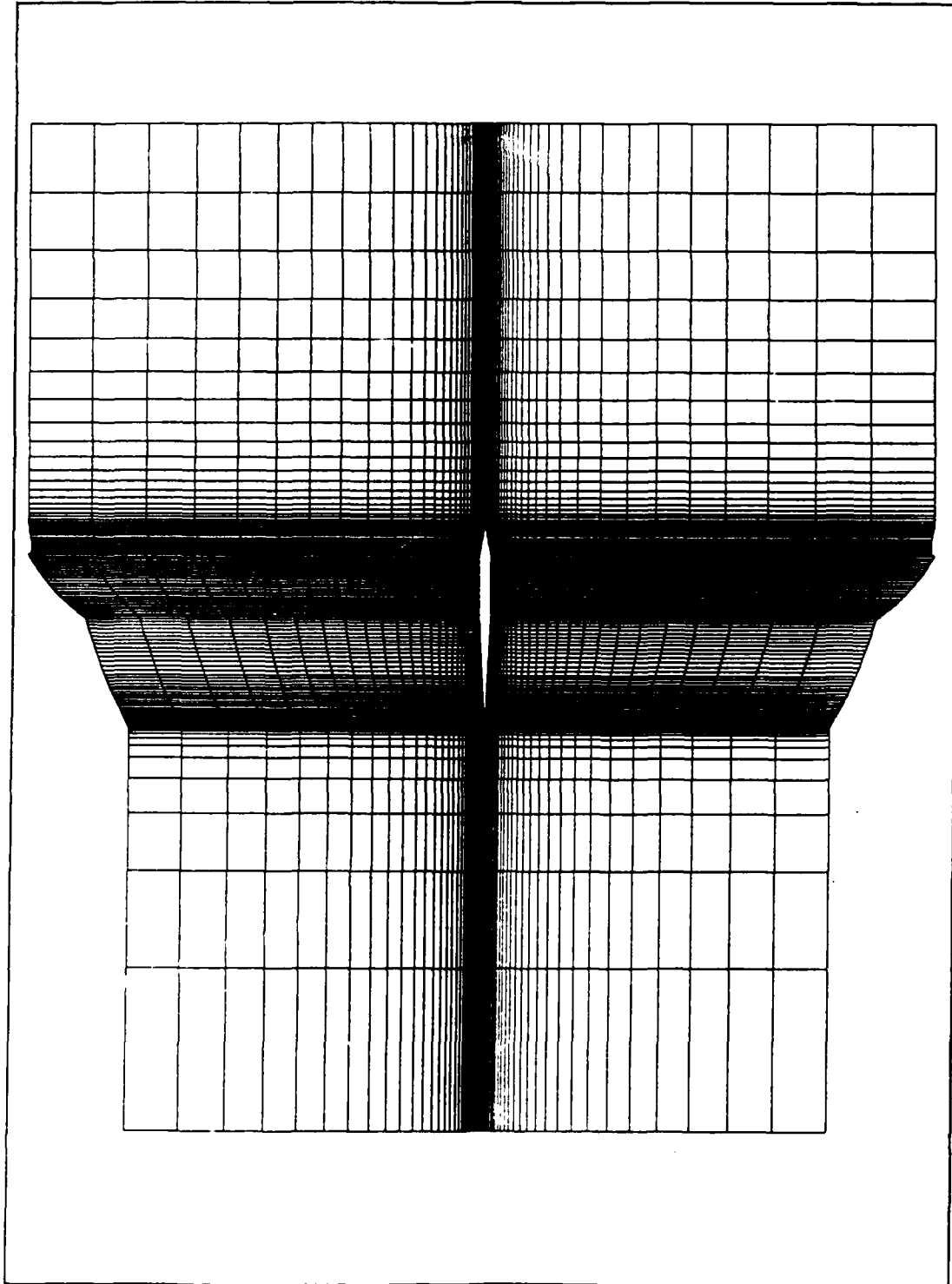
Figure 27.   Cylindrical Grid - side view (130x240x68)

74

Figure 28. Typical Cross-section Upstream of the Apex - front view
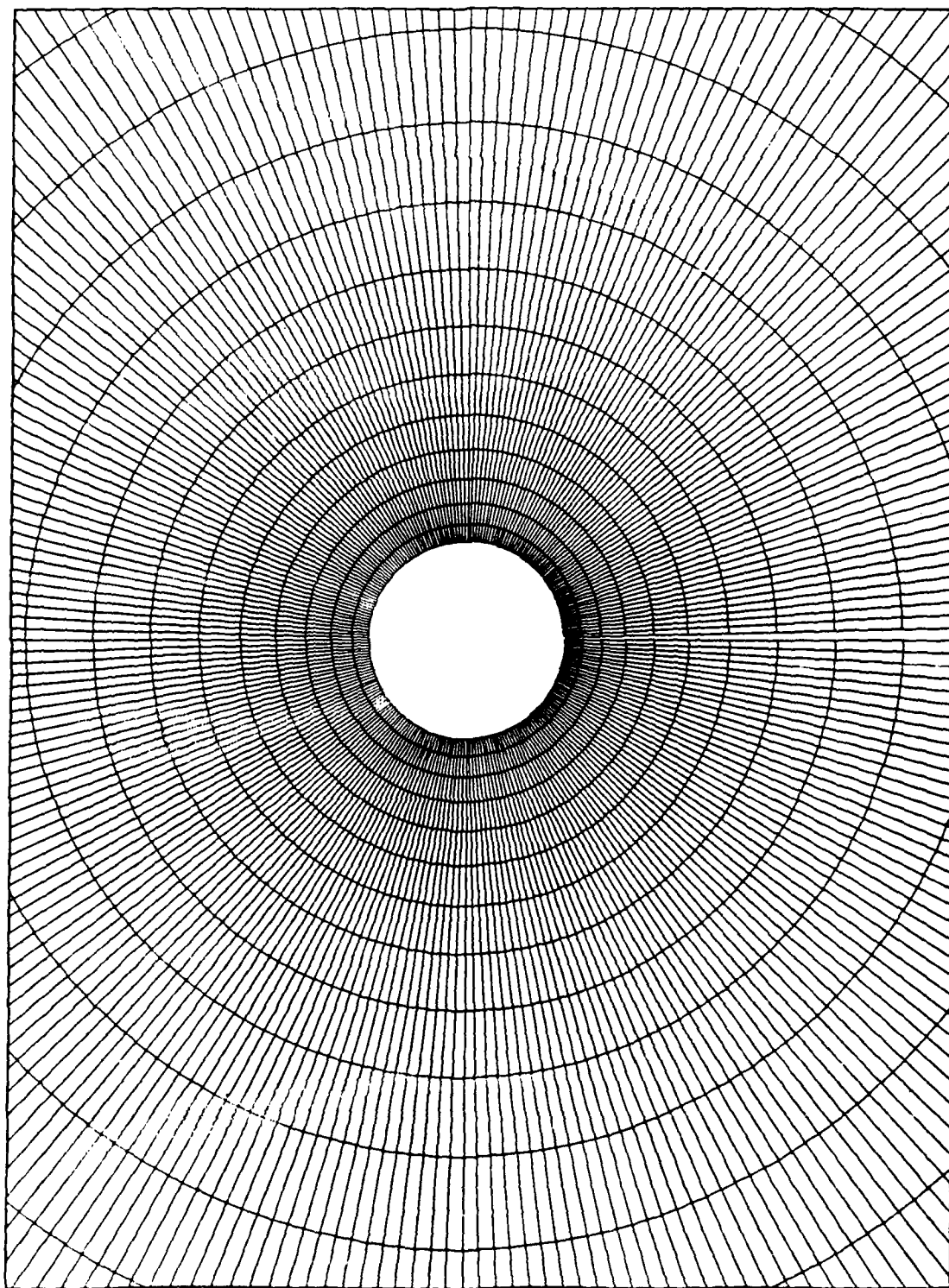
Figure 29.   First Cross-section of the Strake - front view

Figure 30. Typical Cross-section of the Strake - front view
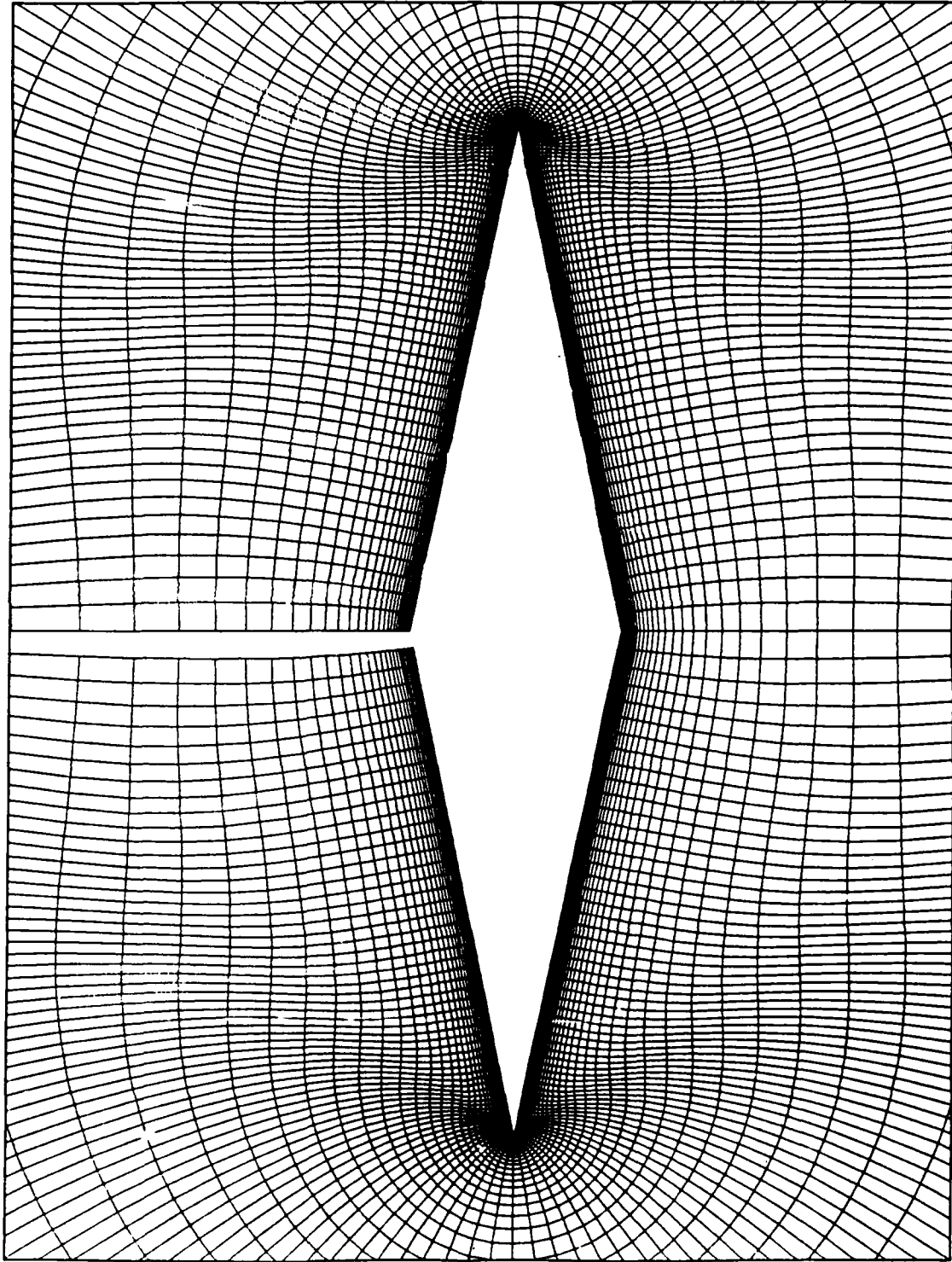
Figure 31. Near Field Grid of the Strake - front view
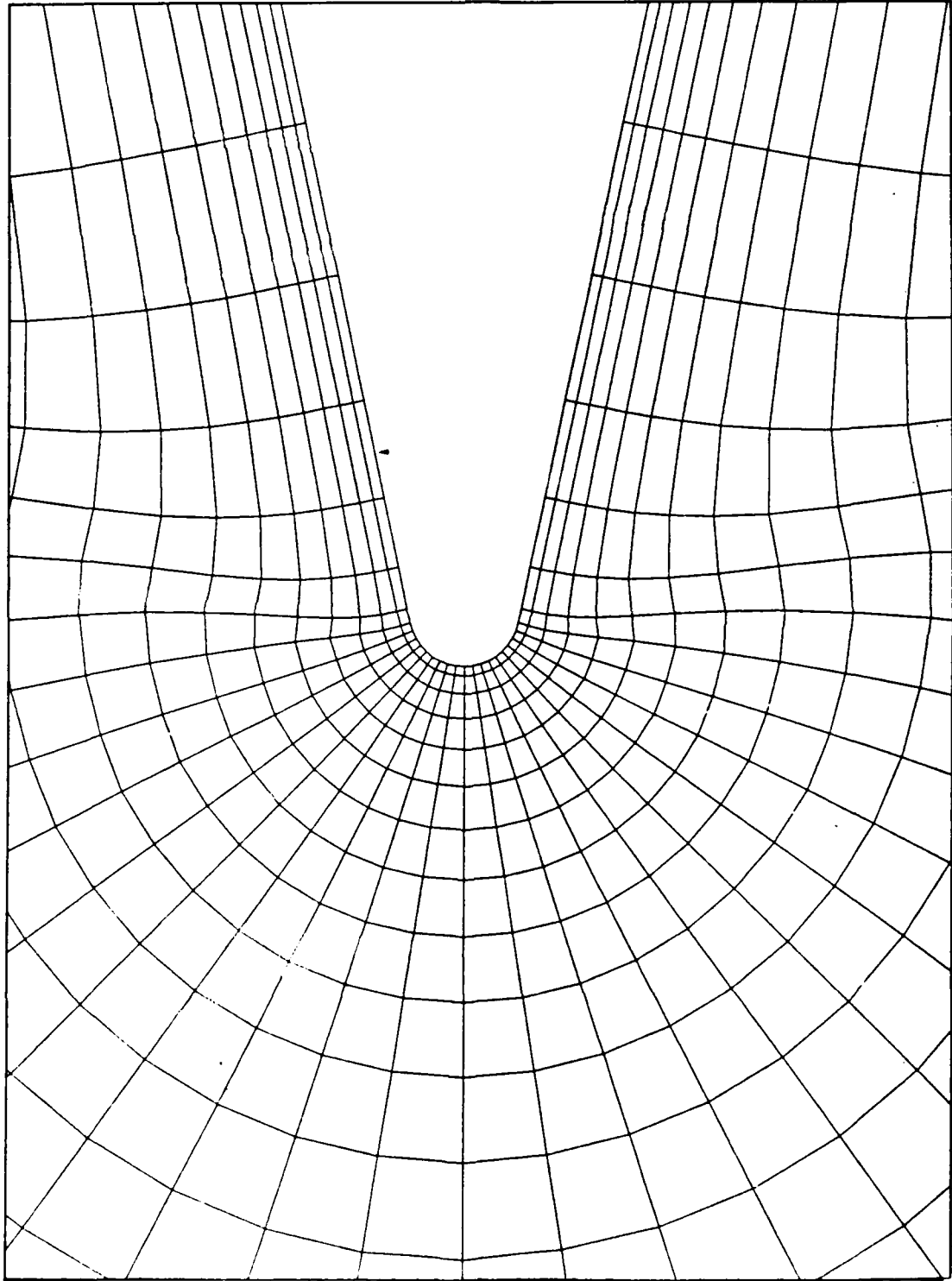
Figure 32.   Leading Edge Detail of the Strake – front view
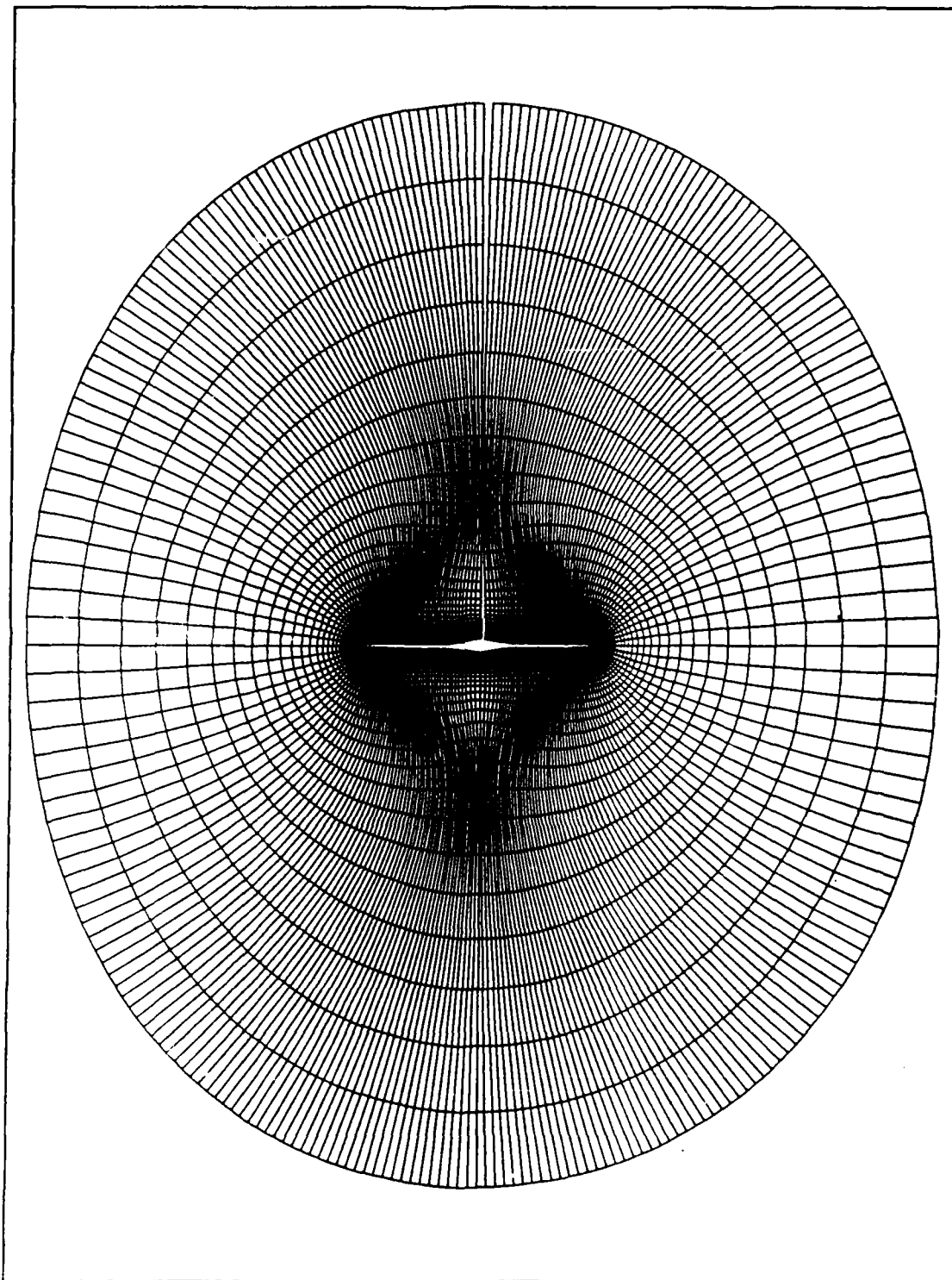
Figure 33.   Typical Cross-section of the Wing - front view
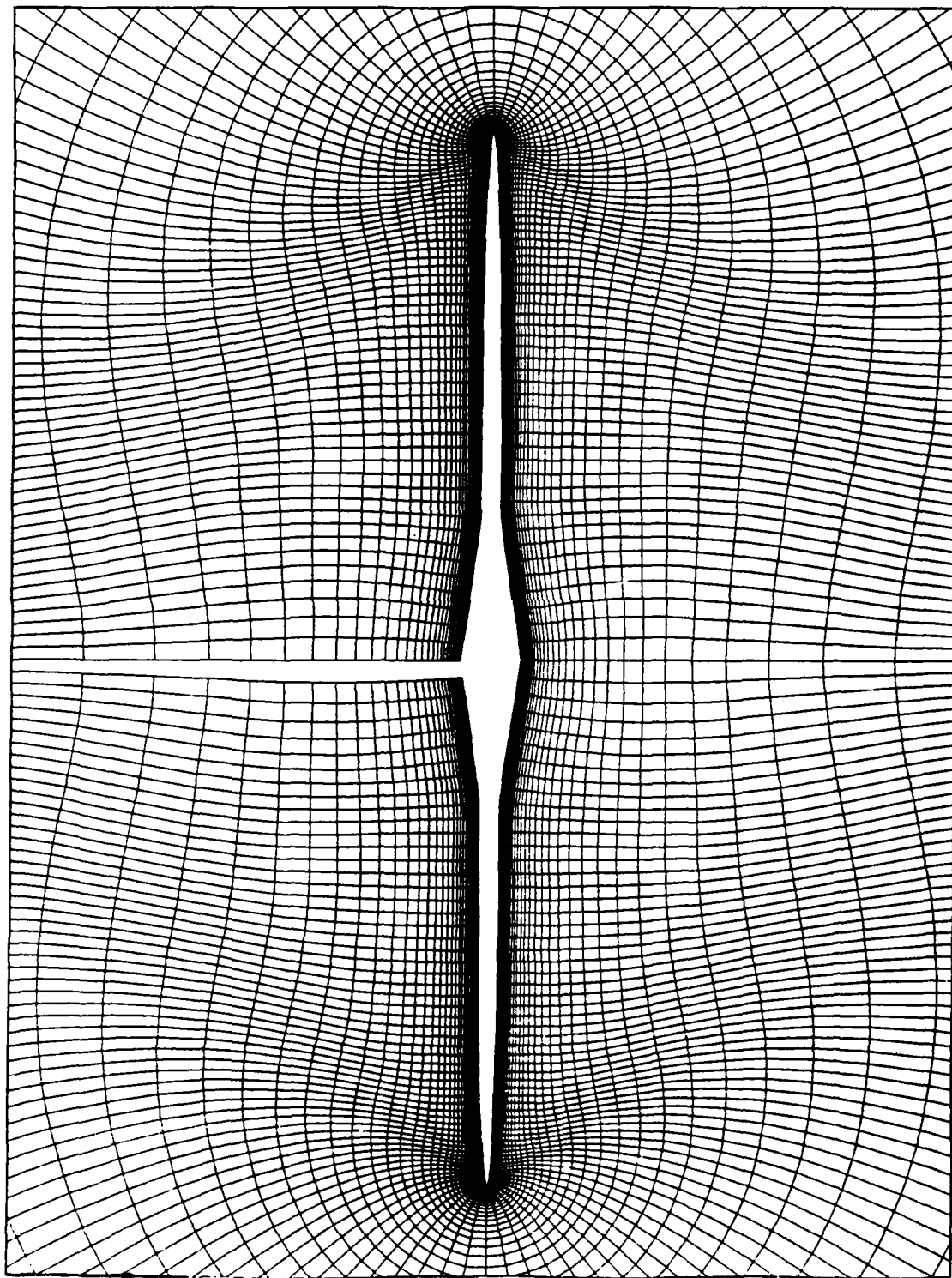
Figure 34. Near Field Grid of the Wing - front view
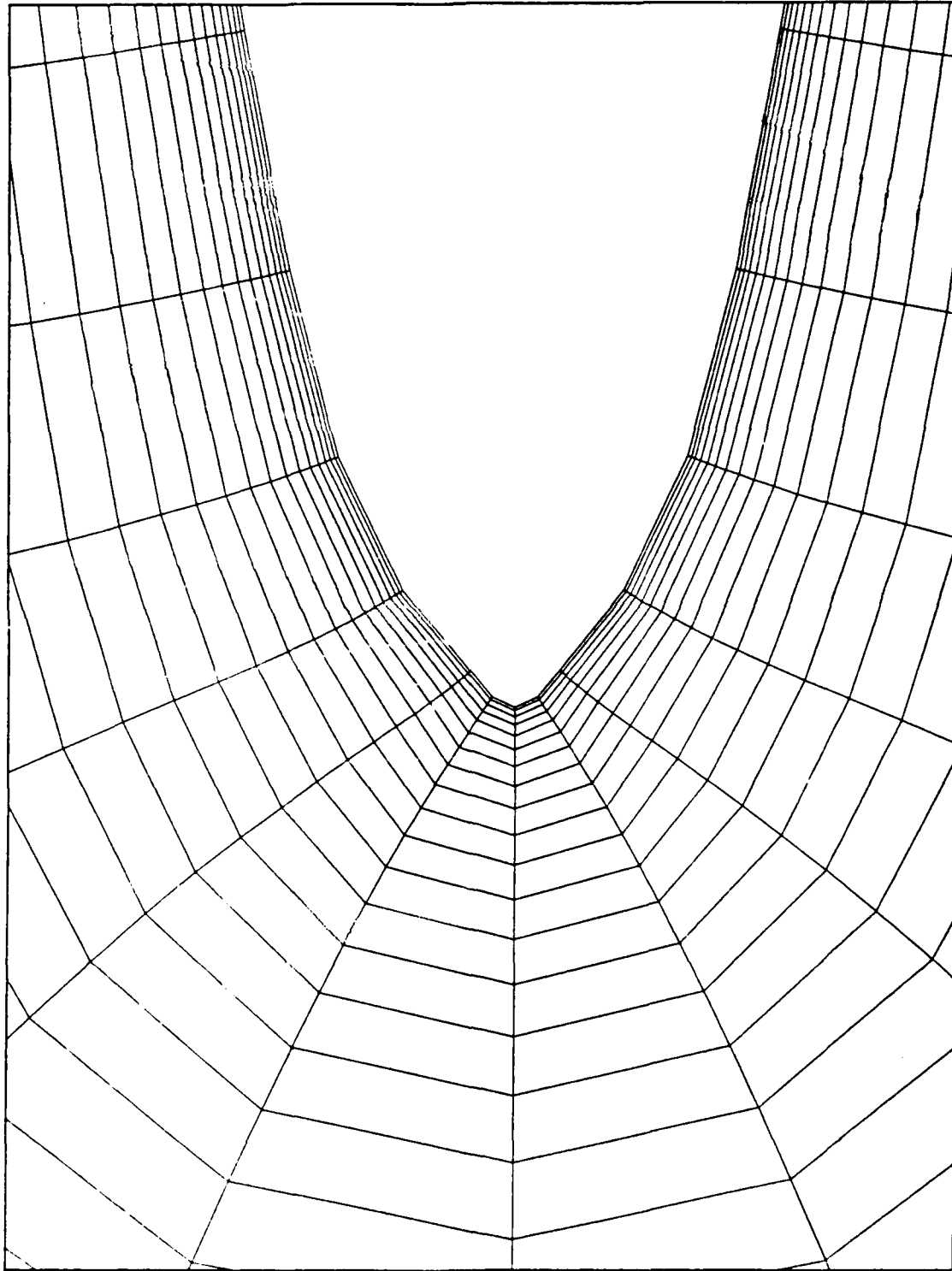
Figure 35. Leading Edge Detail of the Wing - front view
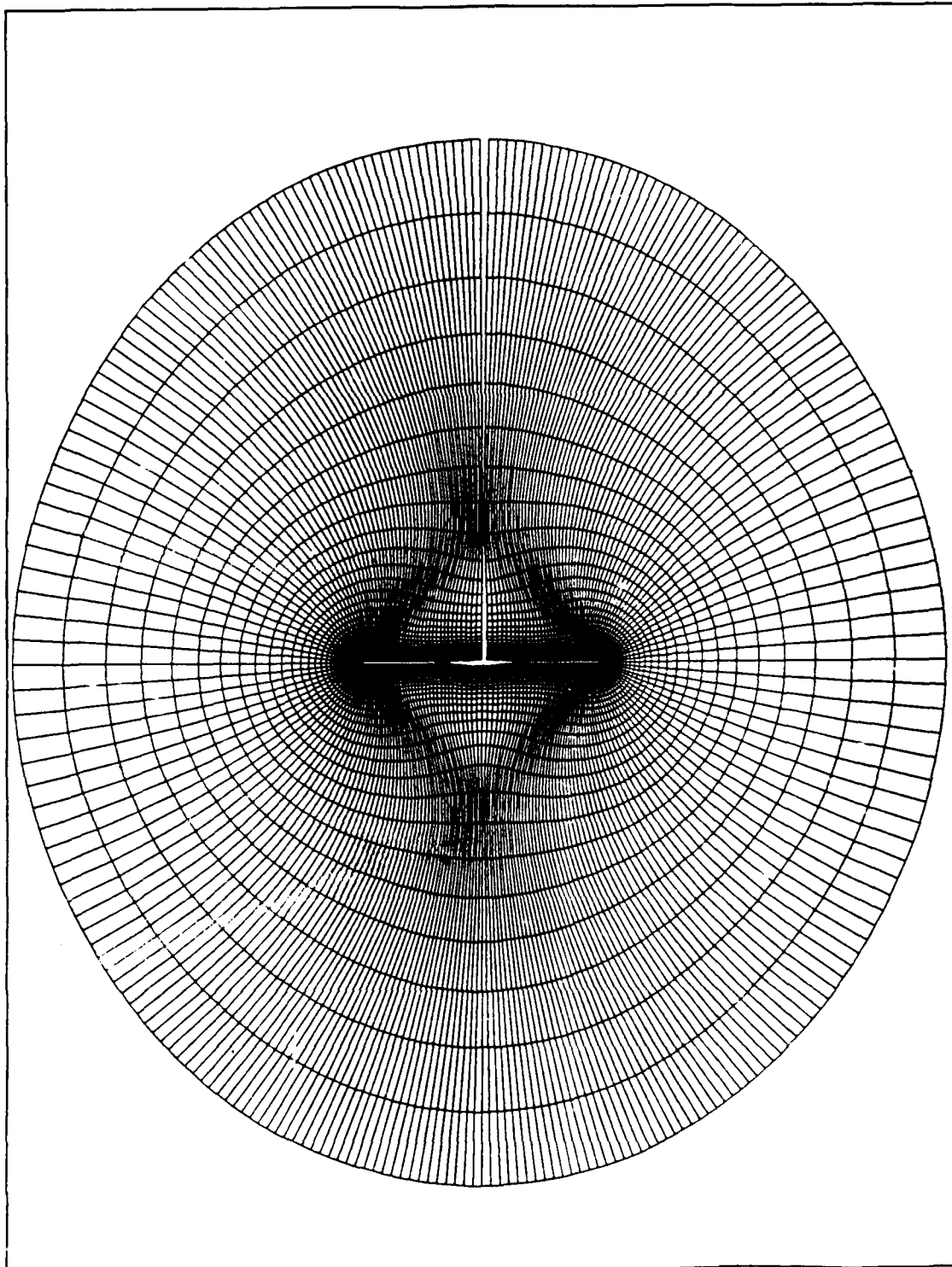
Figure 36.   Typical Cross-section of the Rectangular Section - front view

83
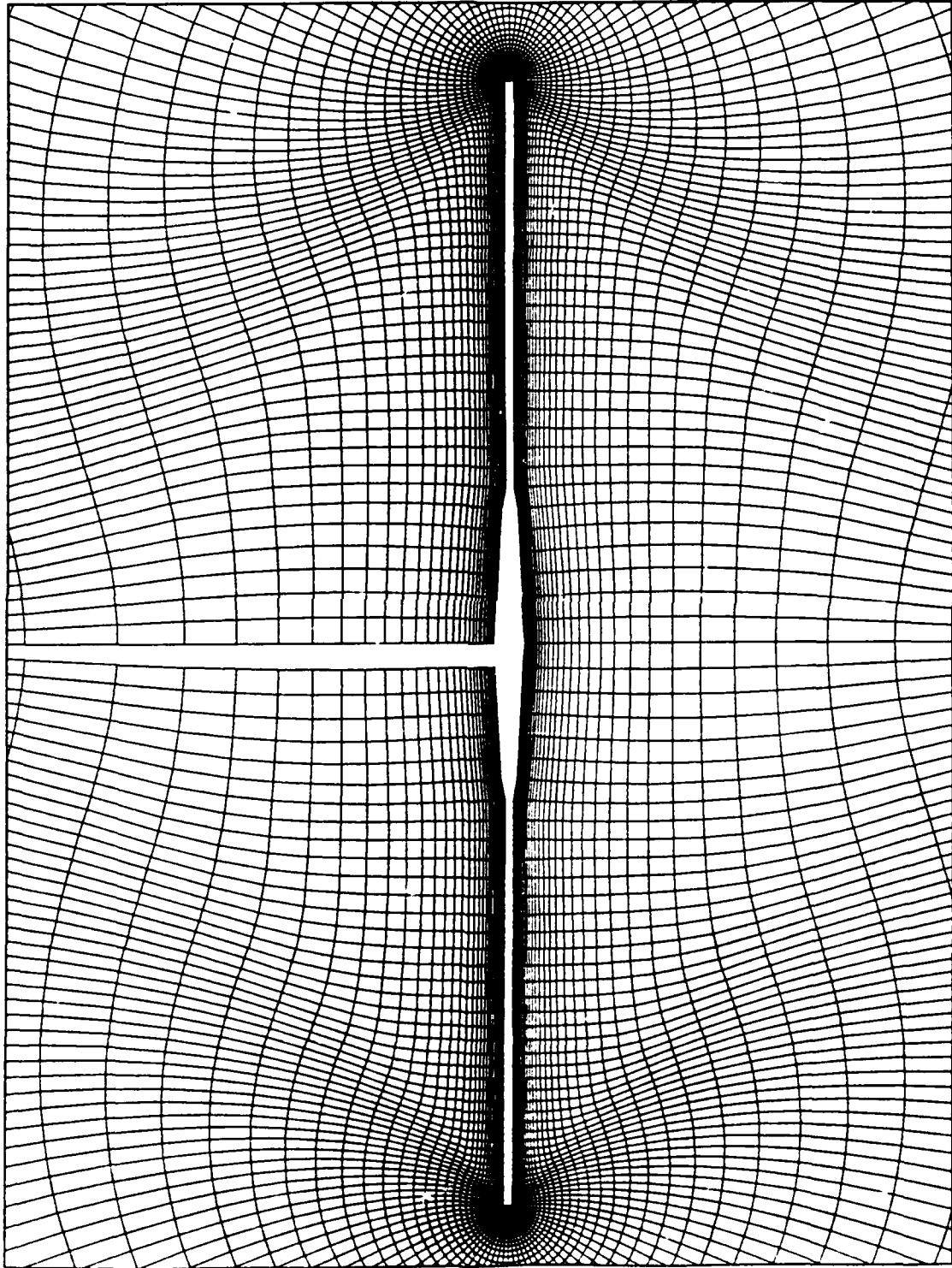
Figure 37. Near Field Grid of the Rectangular Section - front view

84
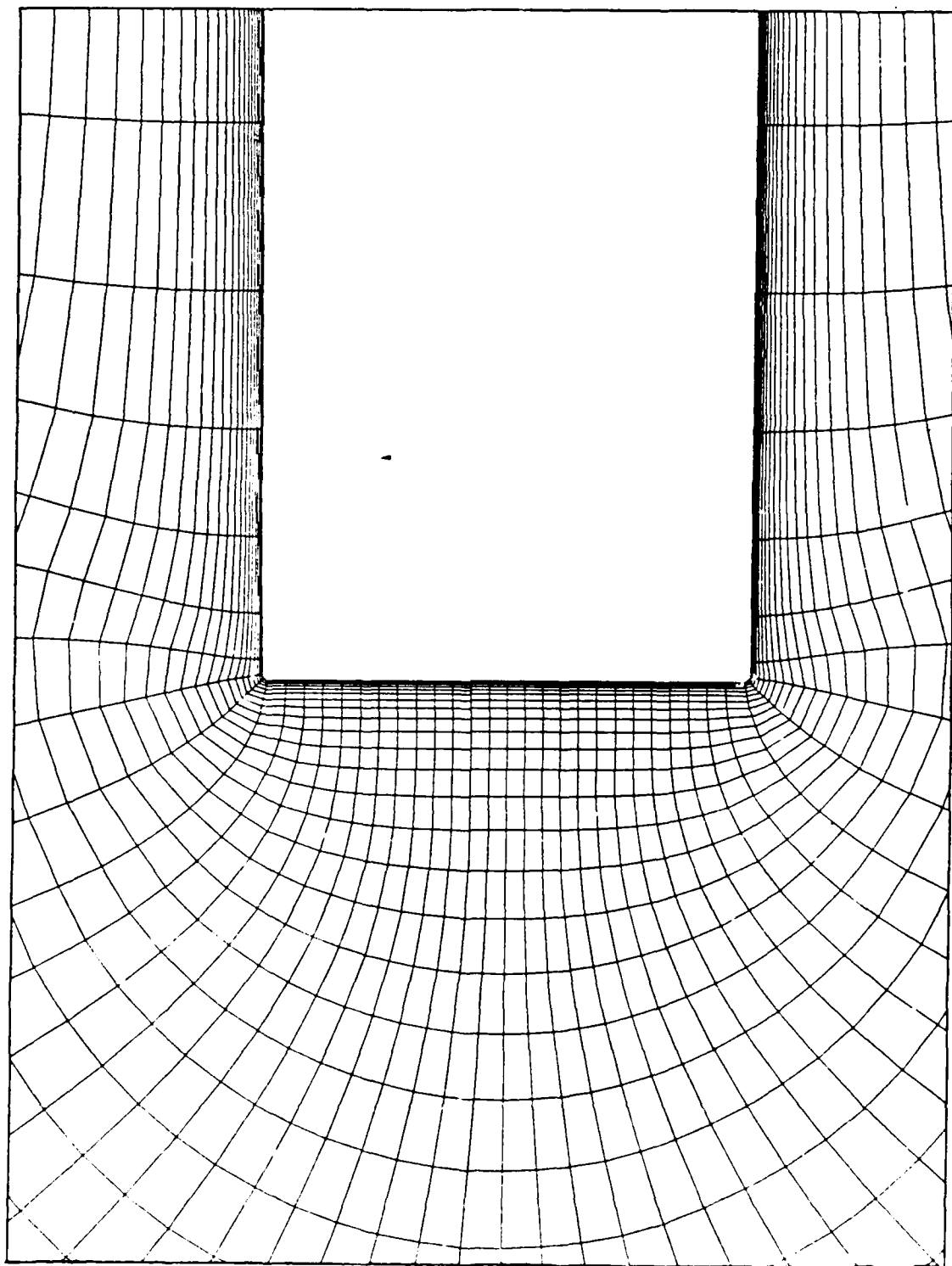
Figure 38. Wing Tip Detail of the Rectangular Section - front view

85
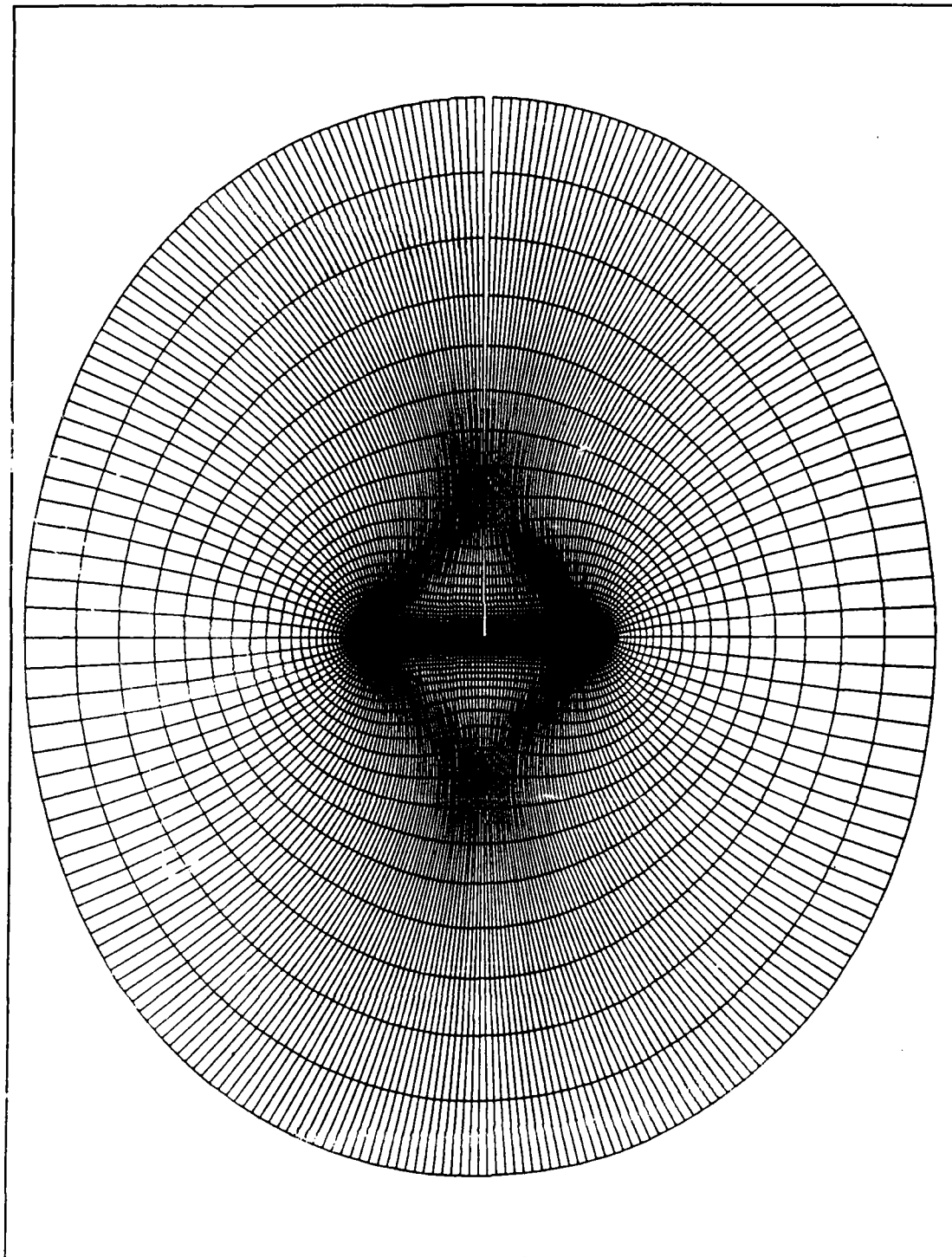
Figure 39.   Typical Cross-section of the Wake - front view
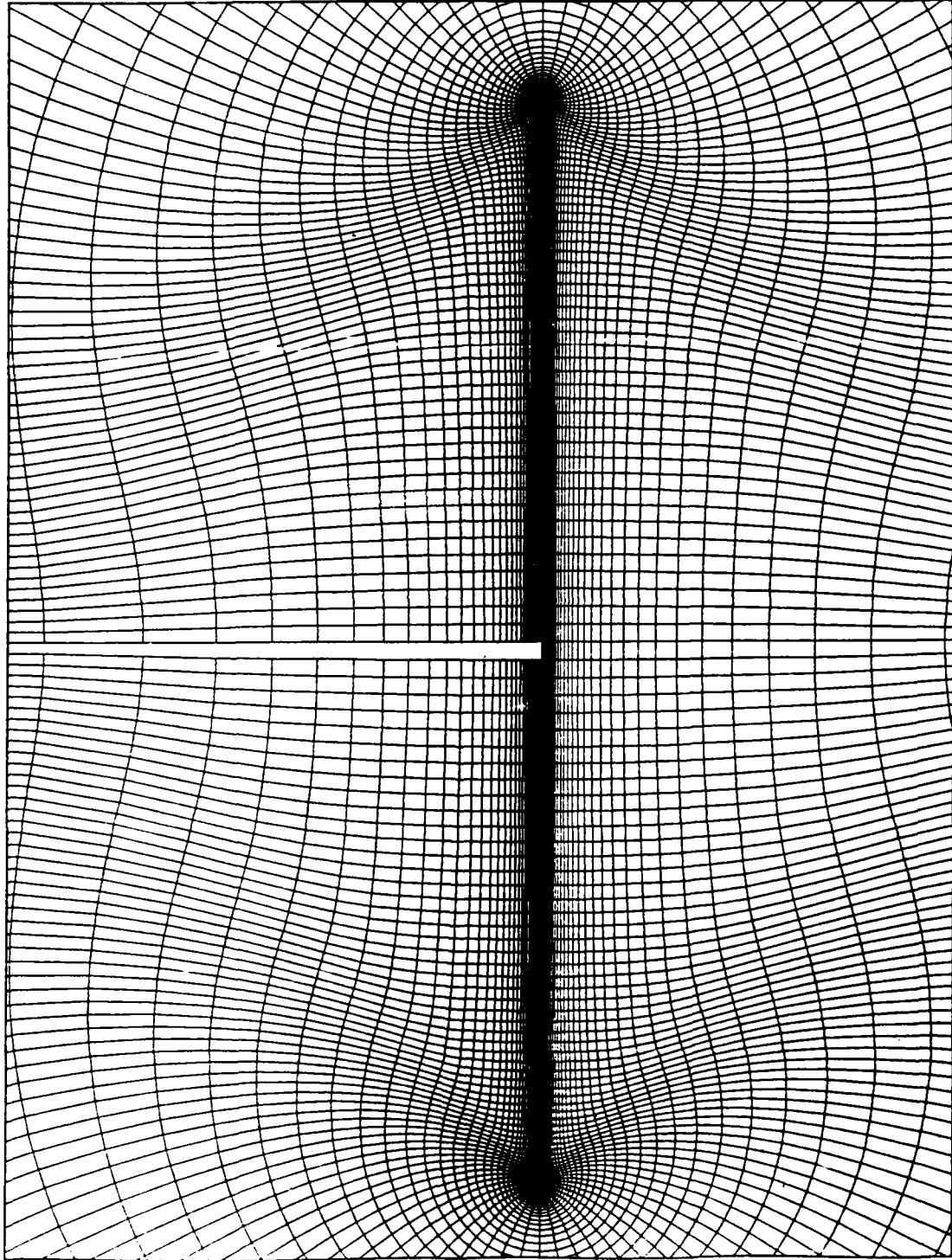
Figure 40.   Near Field Grid of the Wake - front view
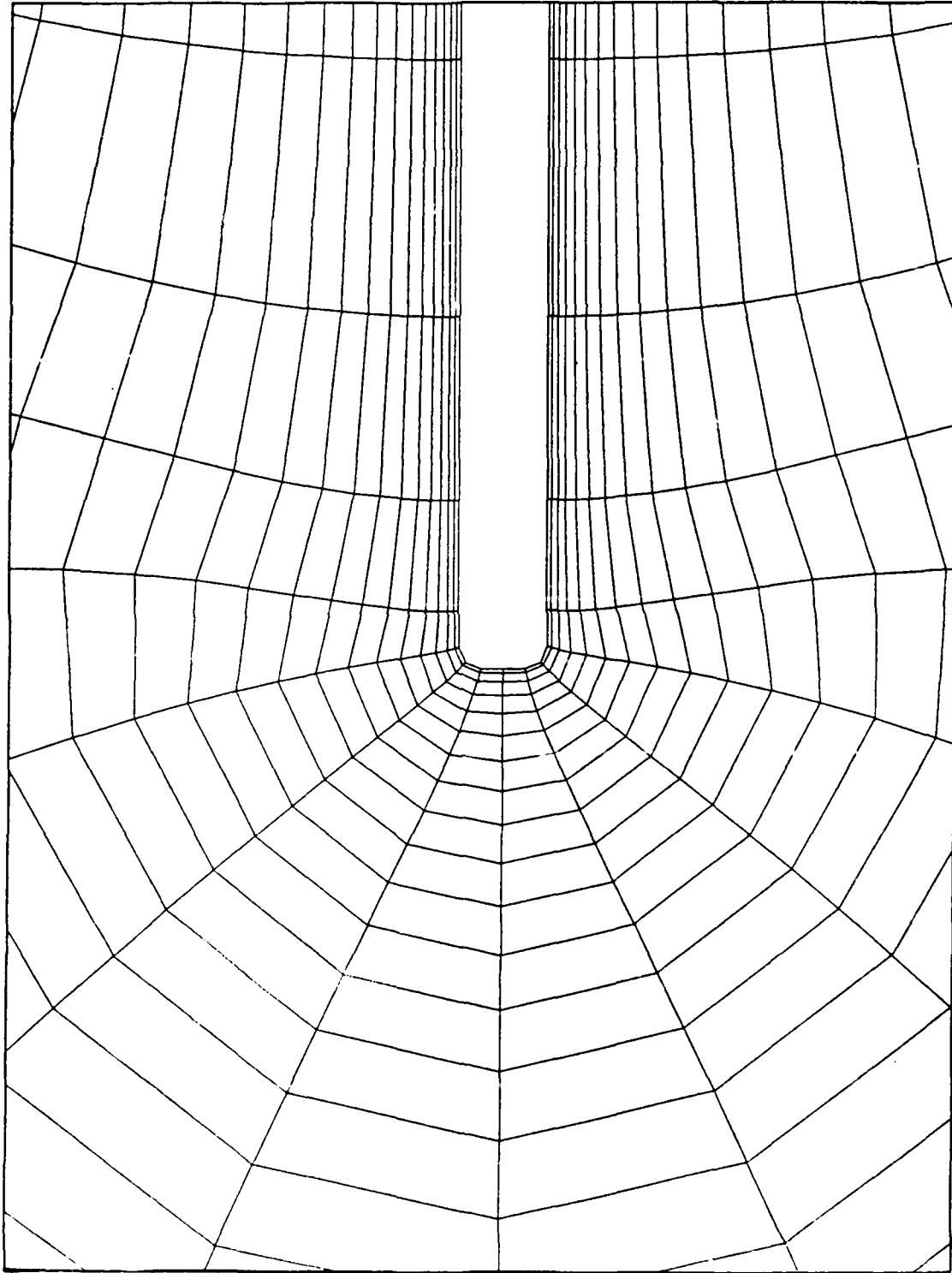
Figure 41. Edge Detail of the Wake - front view

APPENDIX C.   FIELD GRID FIGURES -- SPHERICAL
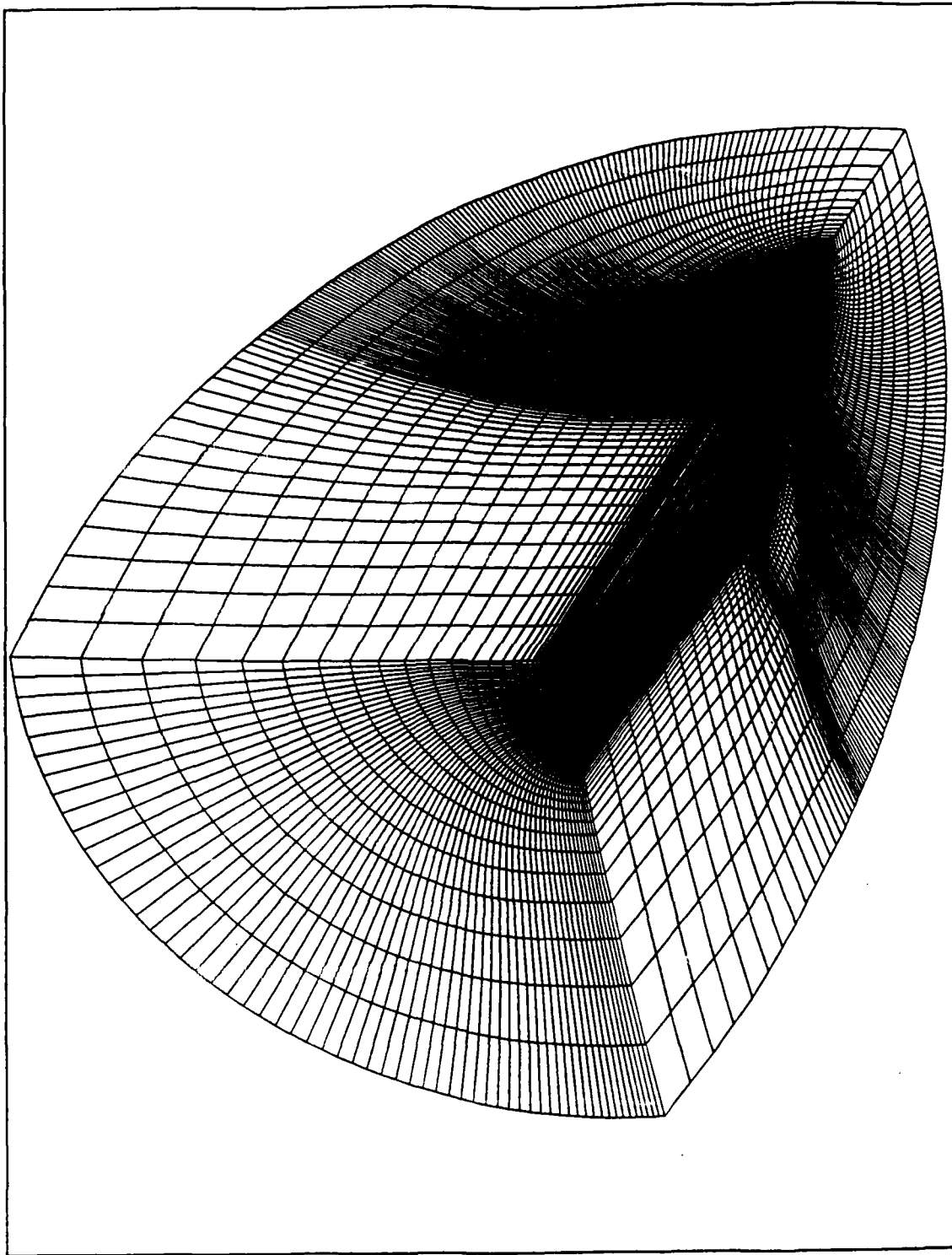
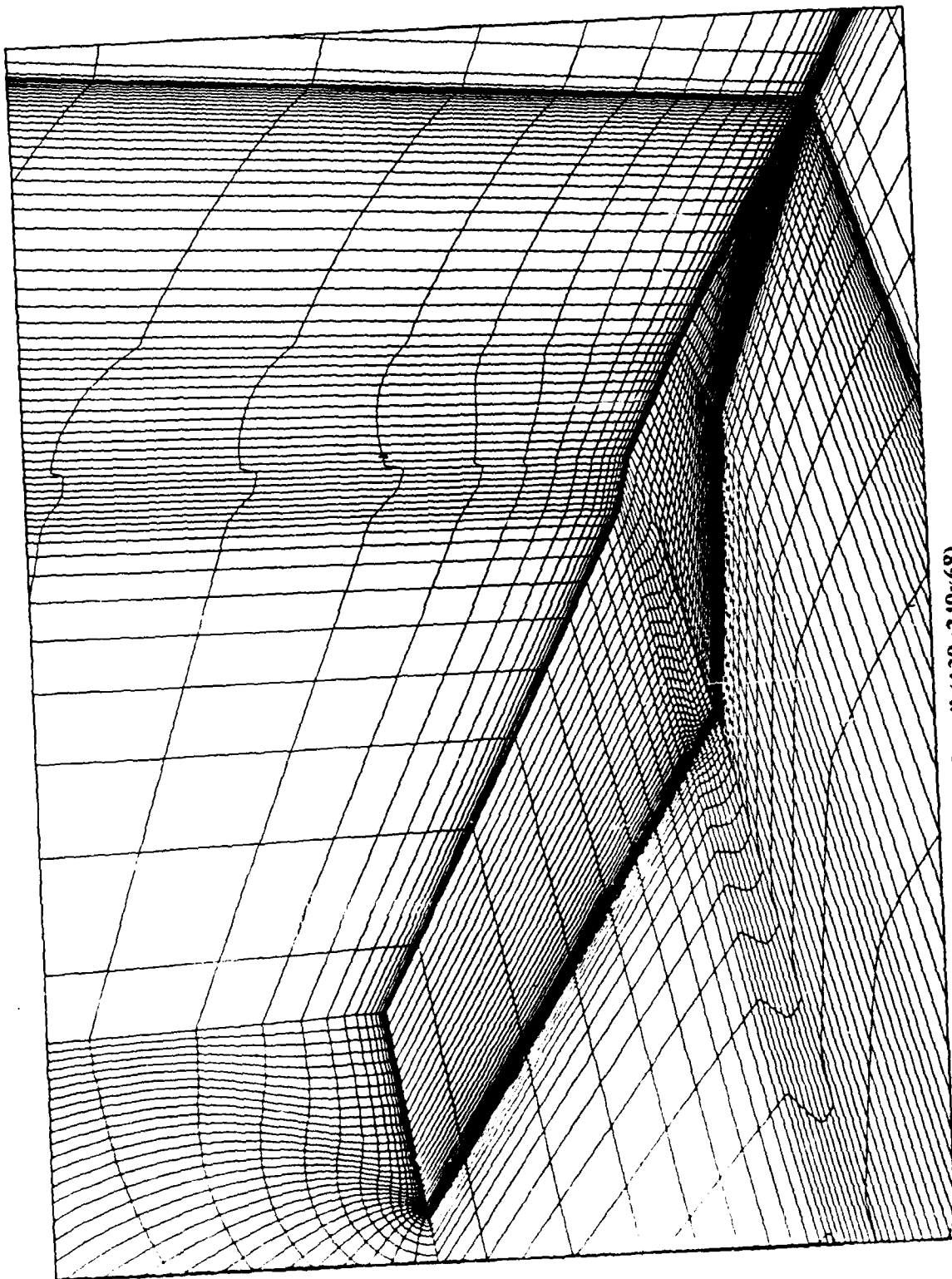Figure 42.   Spherical Grid Configuration (160x240x68)
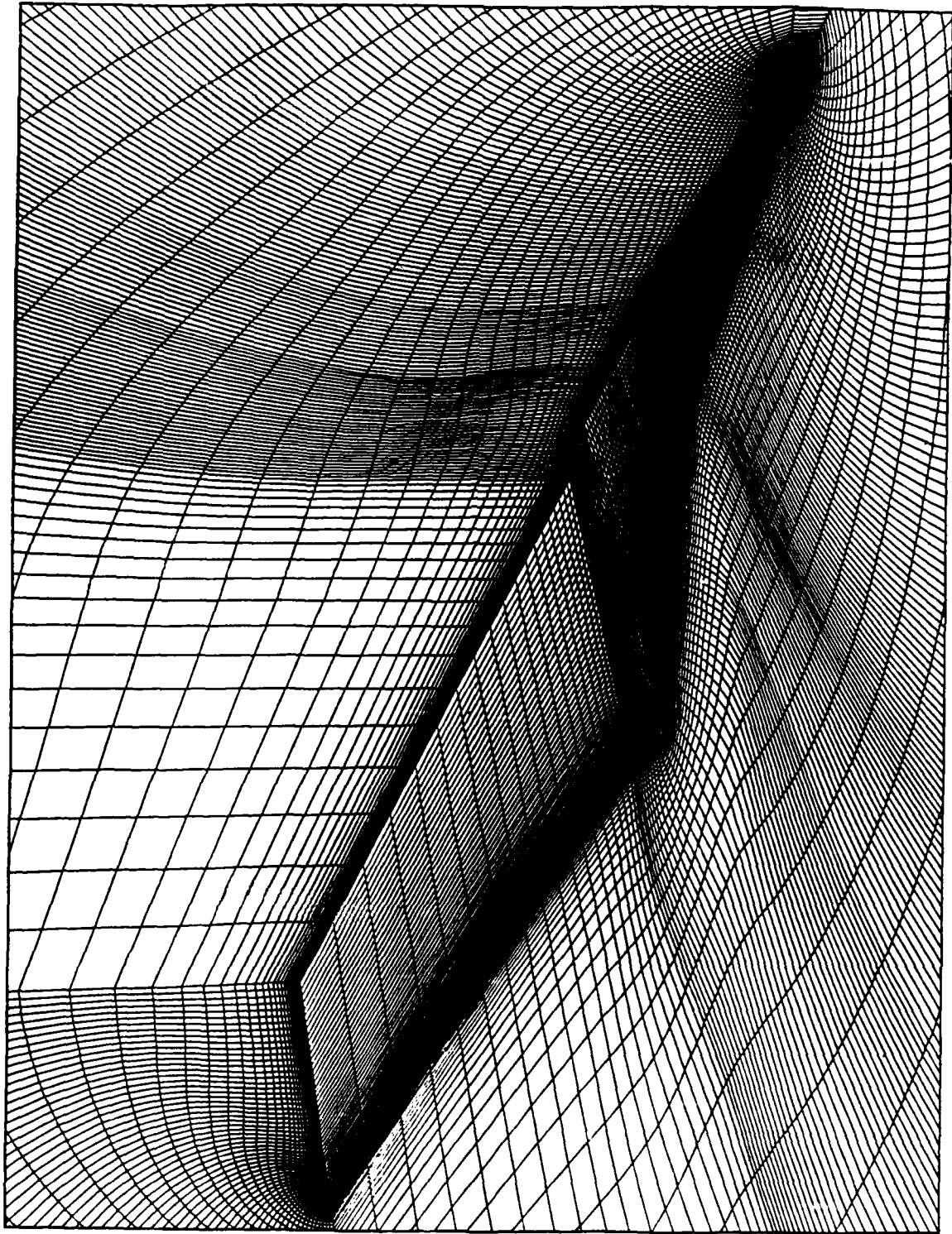
Figure 43. Spherical Grid Configuration Detail (160x240x68)

91
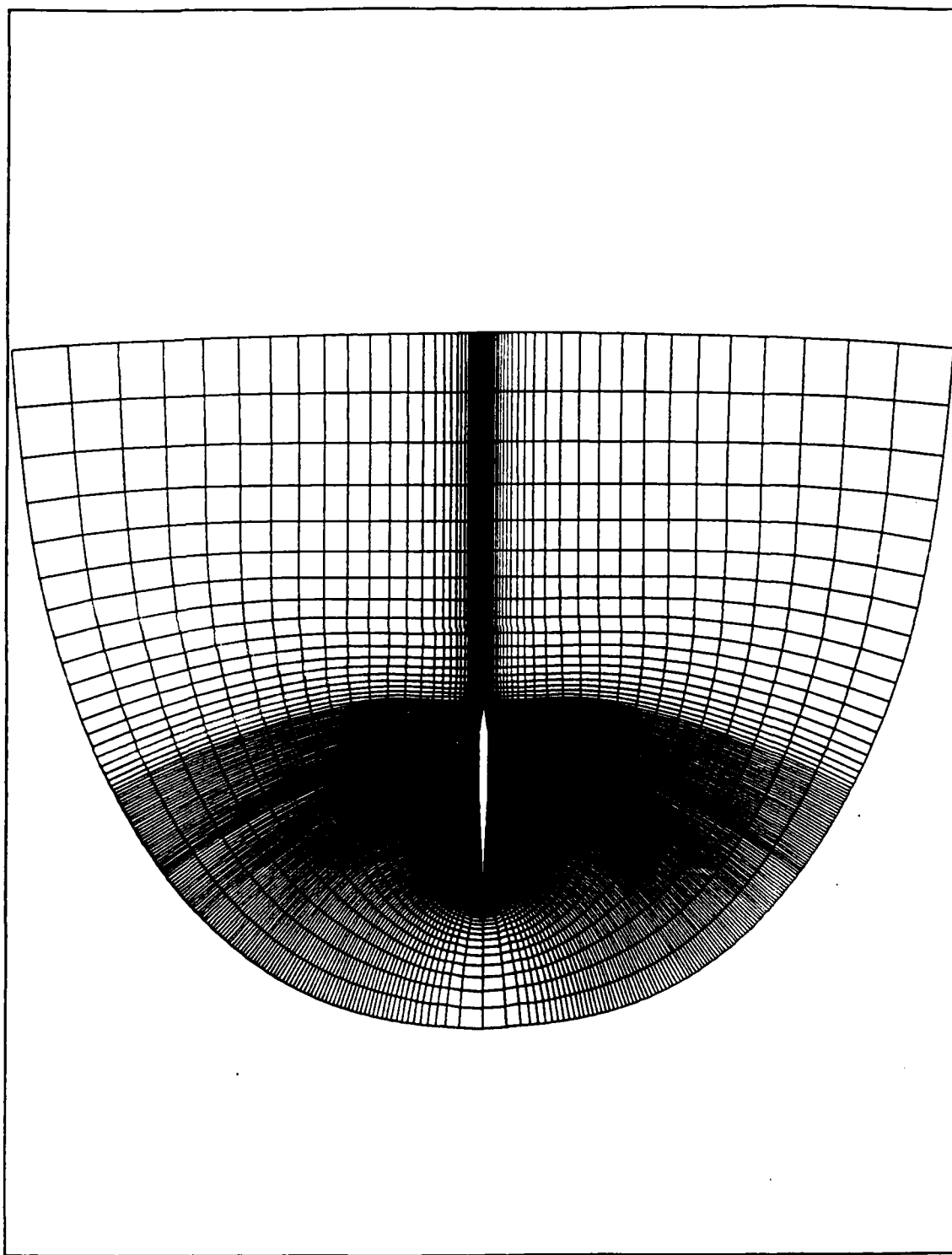
Figure 44. Spherical Grid - side view (160x240x68)
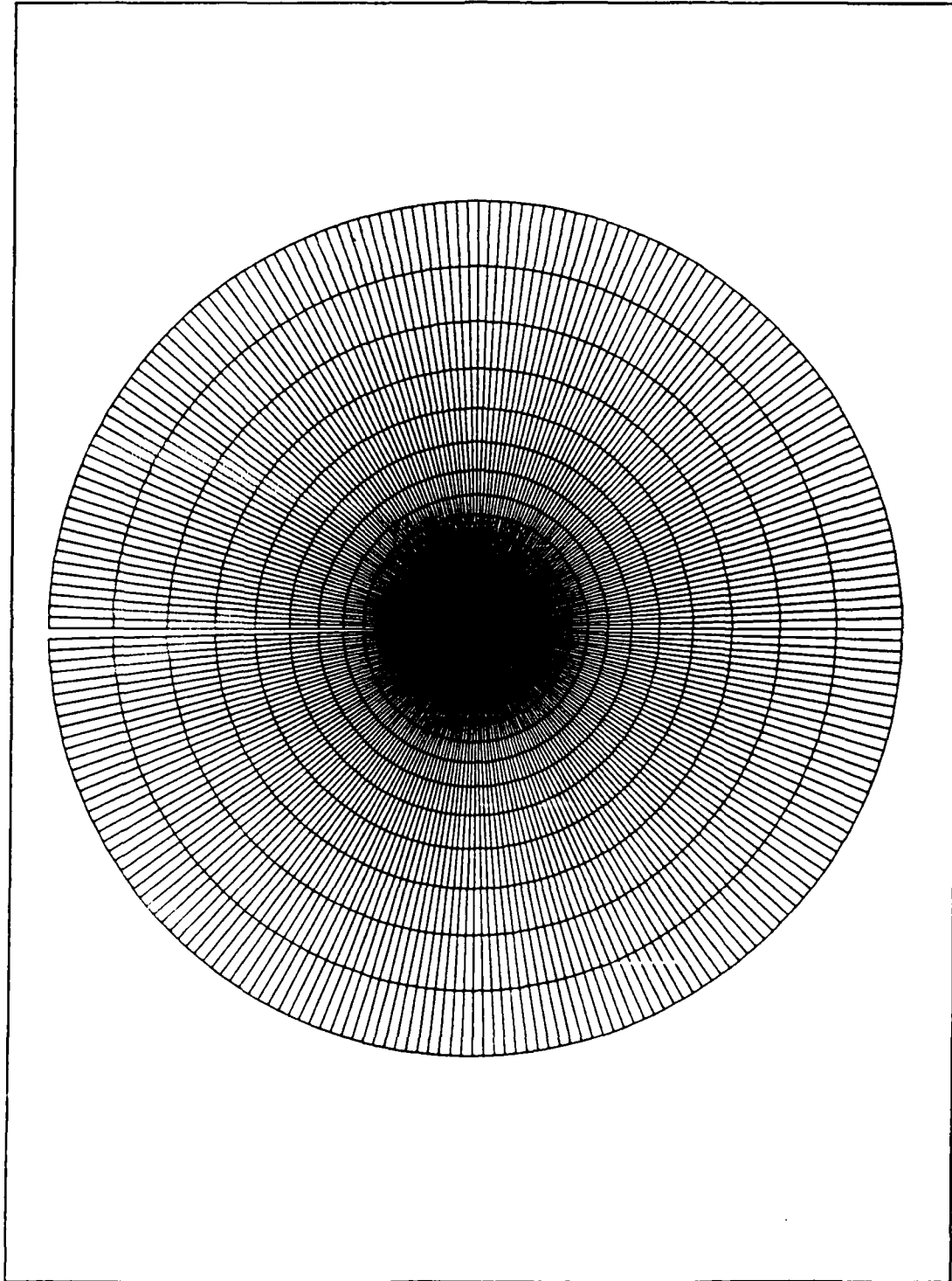
92
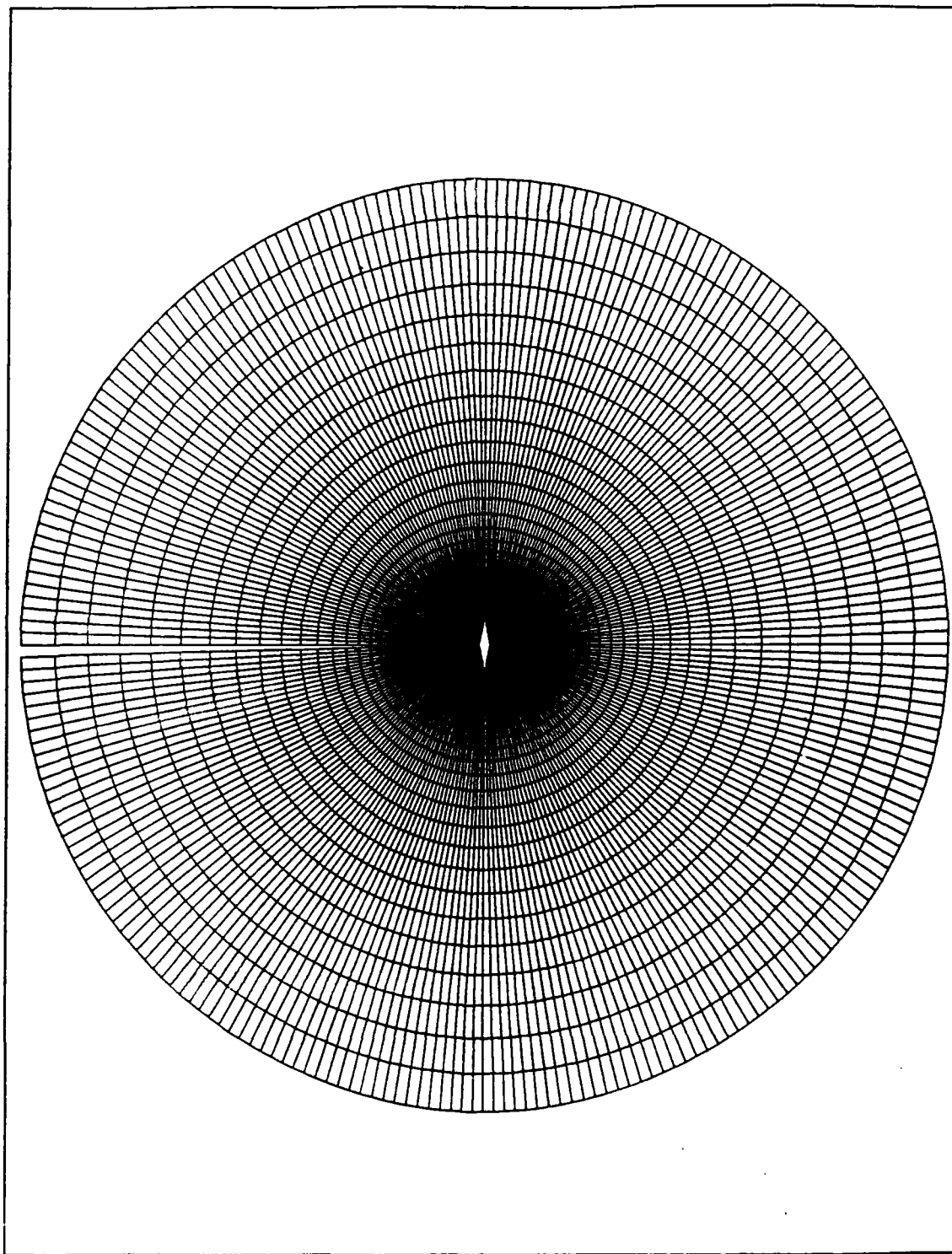
Figure 45.   Typical Cross-section of the strake - front view
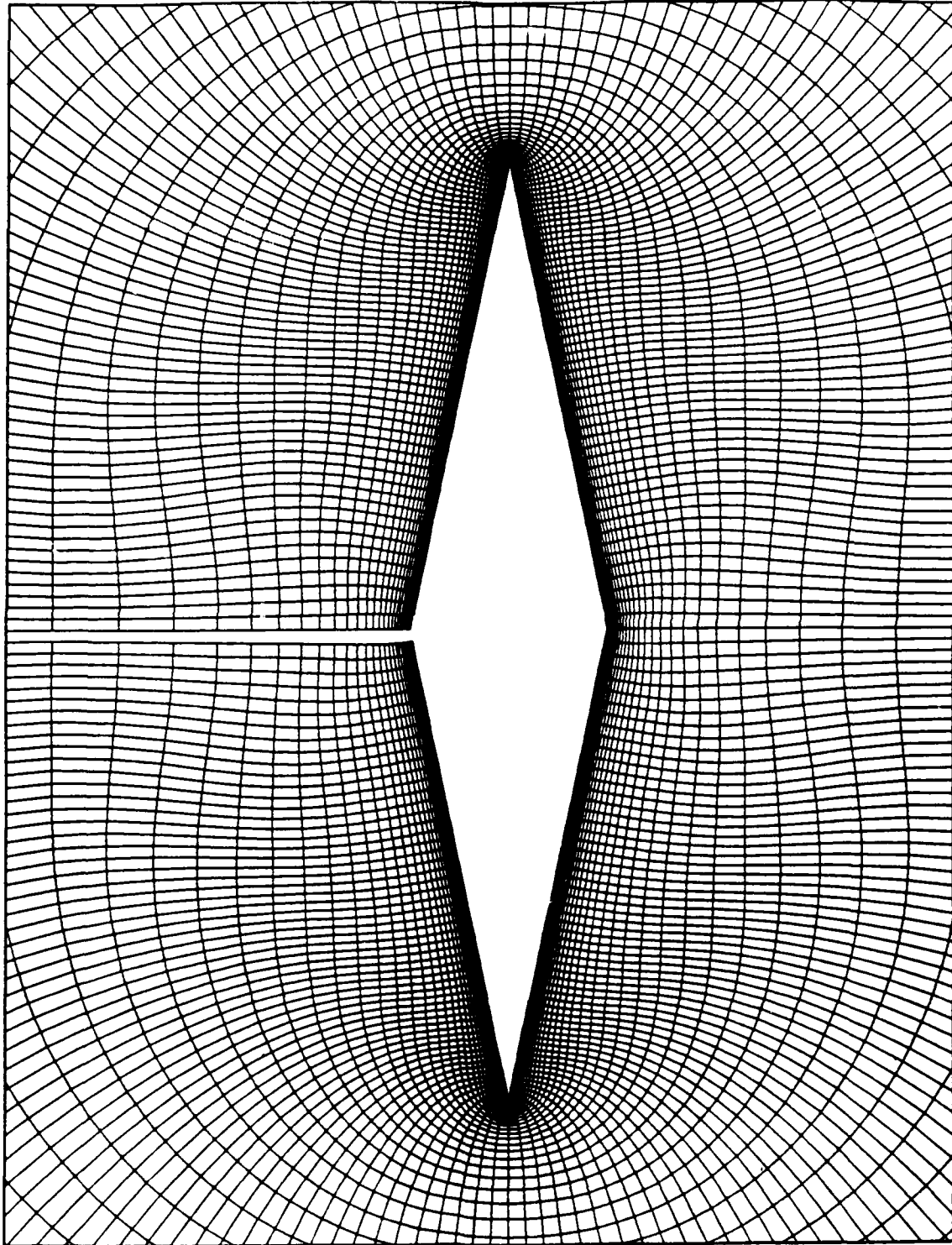
Figure 46. Near Field Grid of the strake - front view
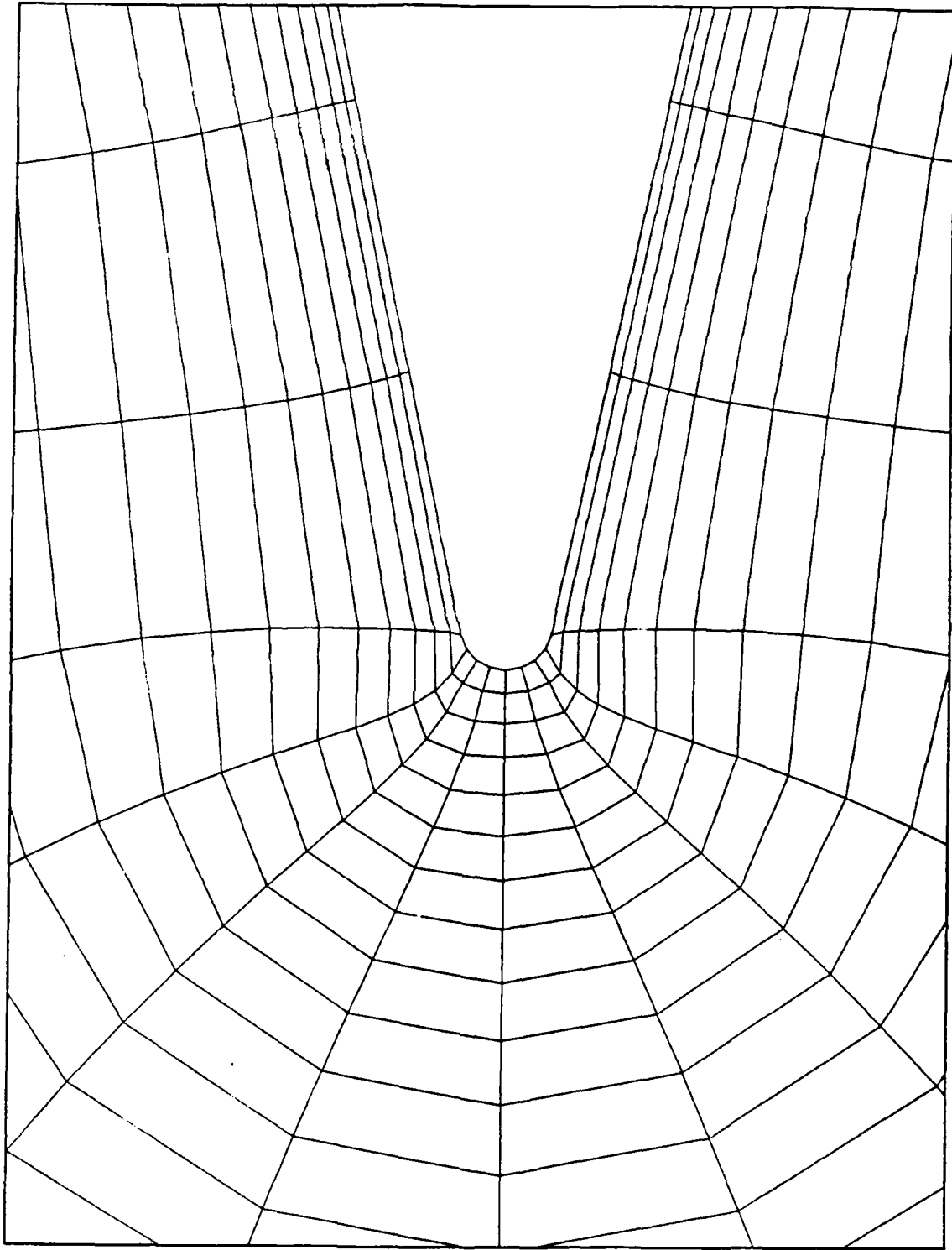
Figure 47. Leading Edge Detail of the Strake - front view
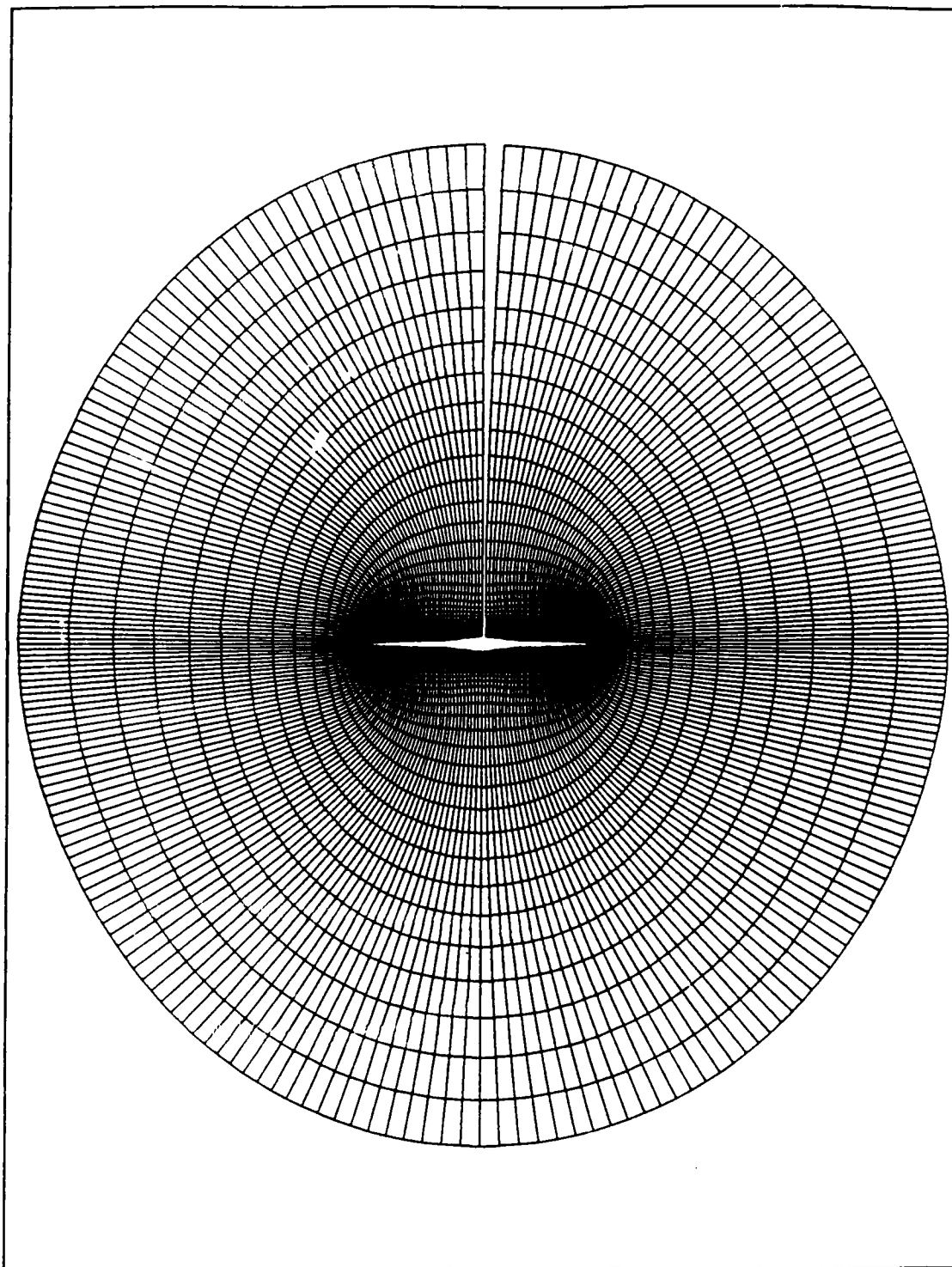
Figure 48.   Typical Cross-section of the Wing - front view
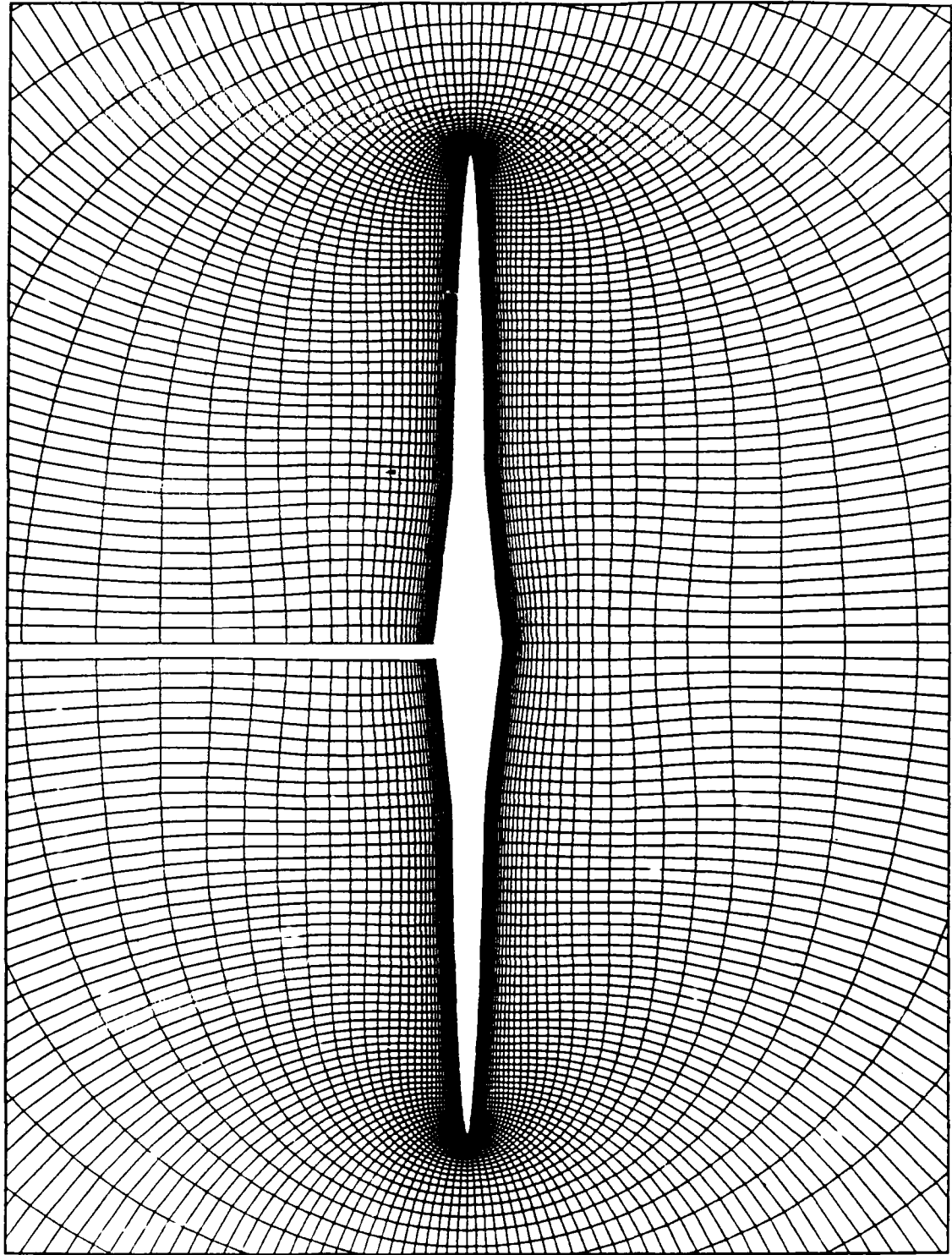
Figure 49. Near Field Grid of the Wing - front view
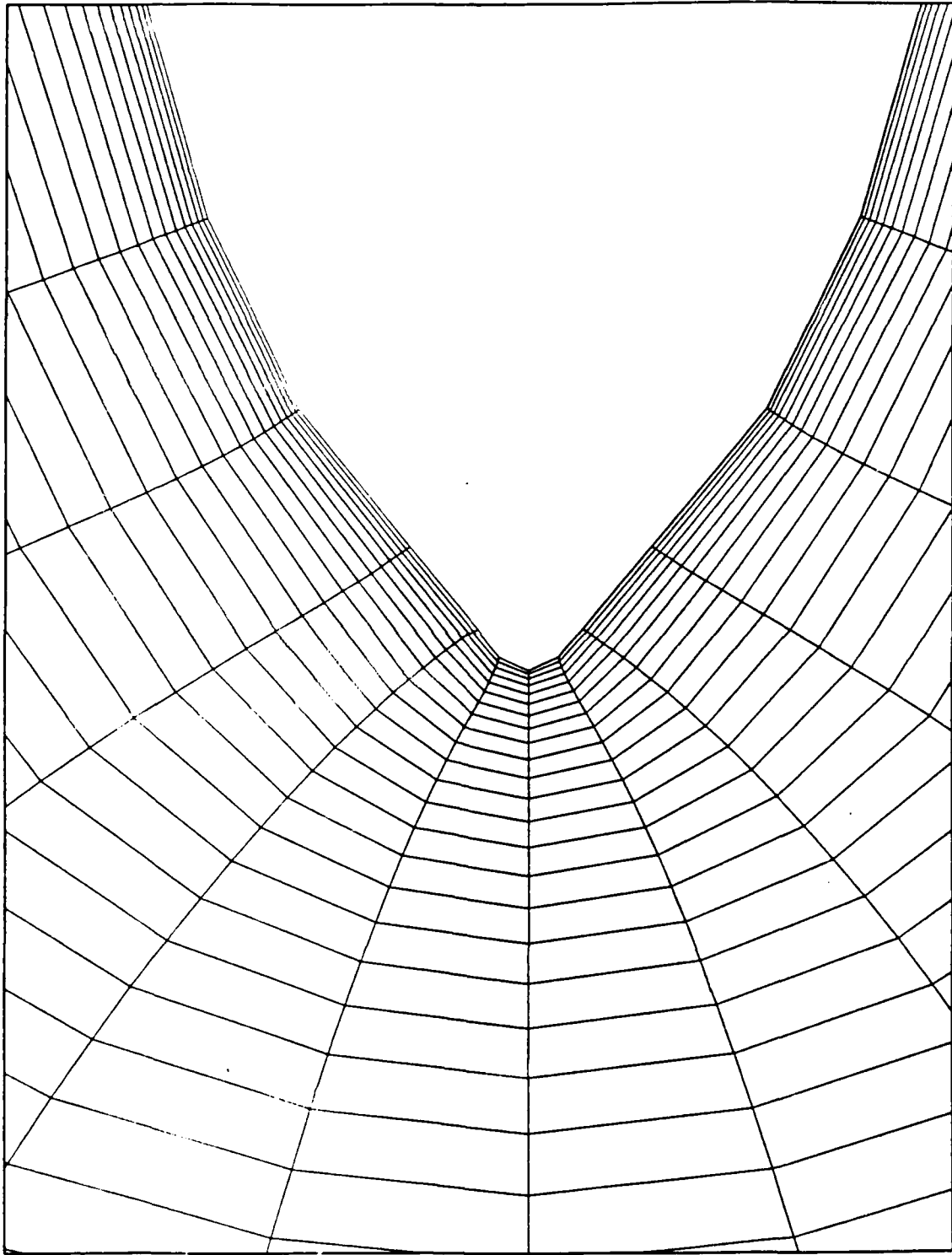
Figure 50.   Leading Edge Detail of the Wing - front view
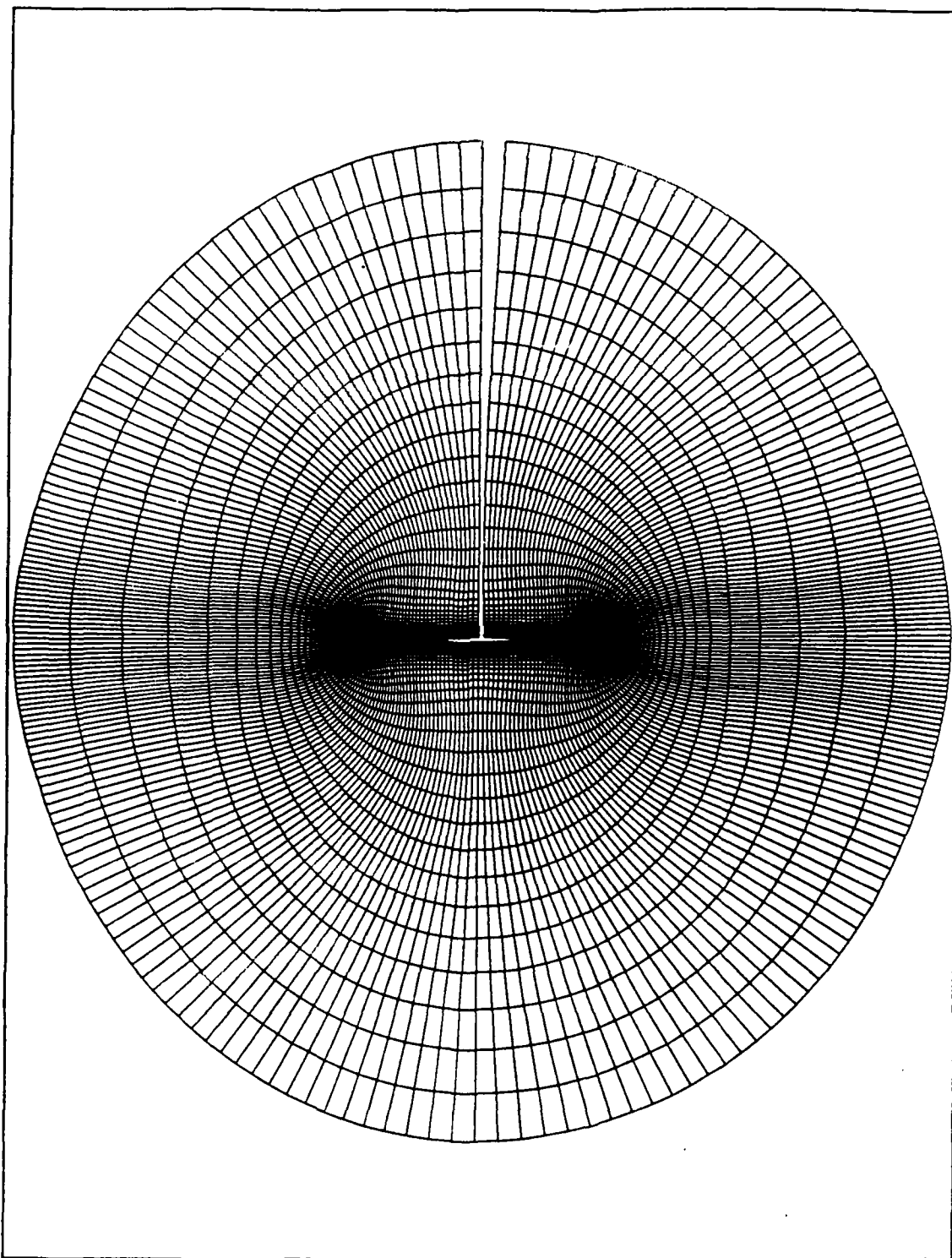
Figure 51.   Typical Cross-section of the Rectangular Section - front view
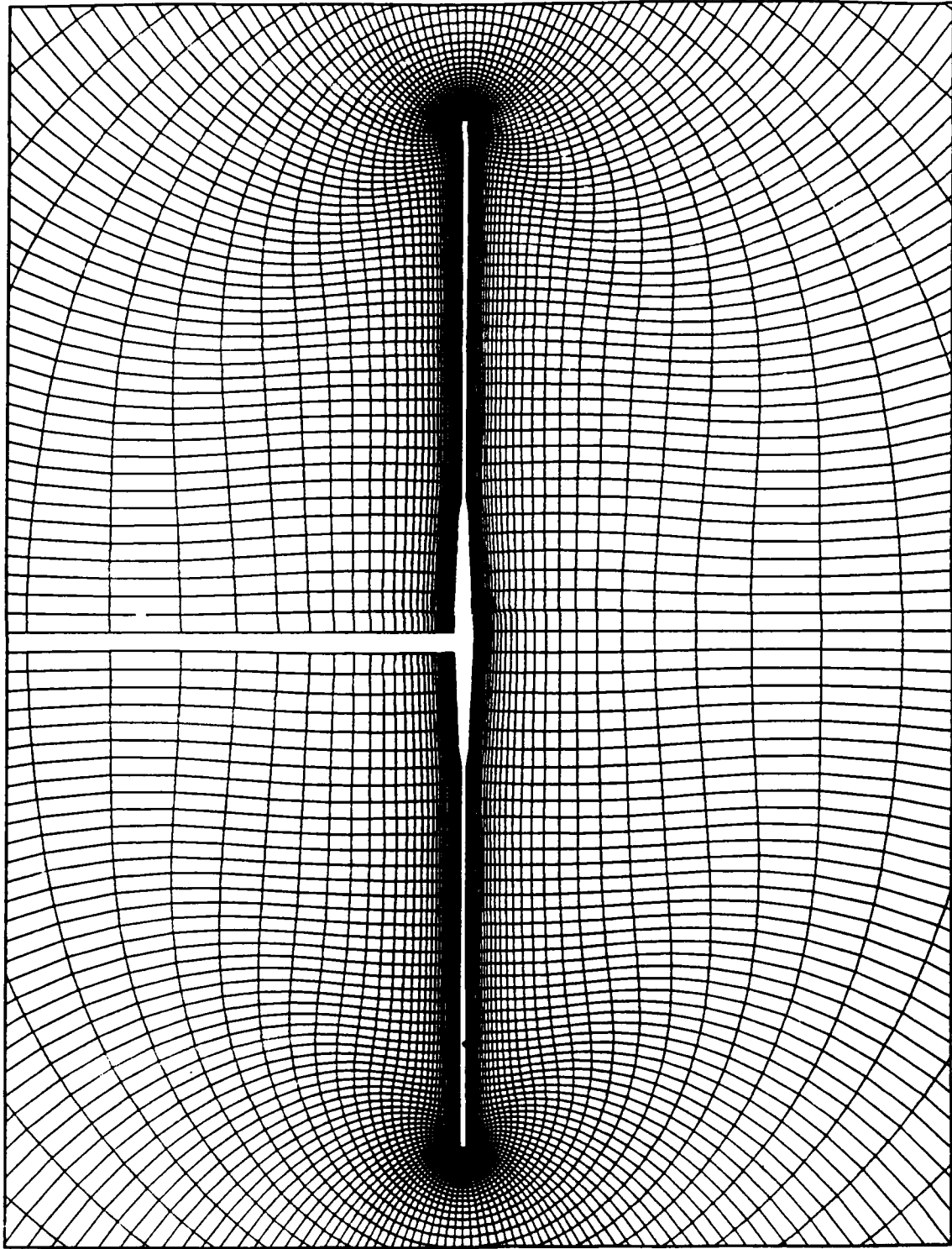
Figure 52. Near Field Grid of the Rectangular Section- front view
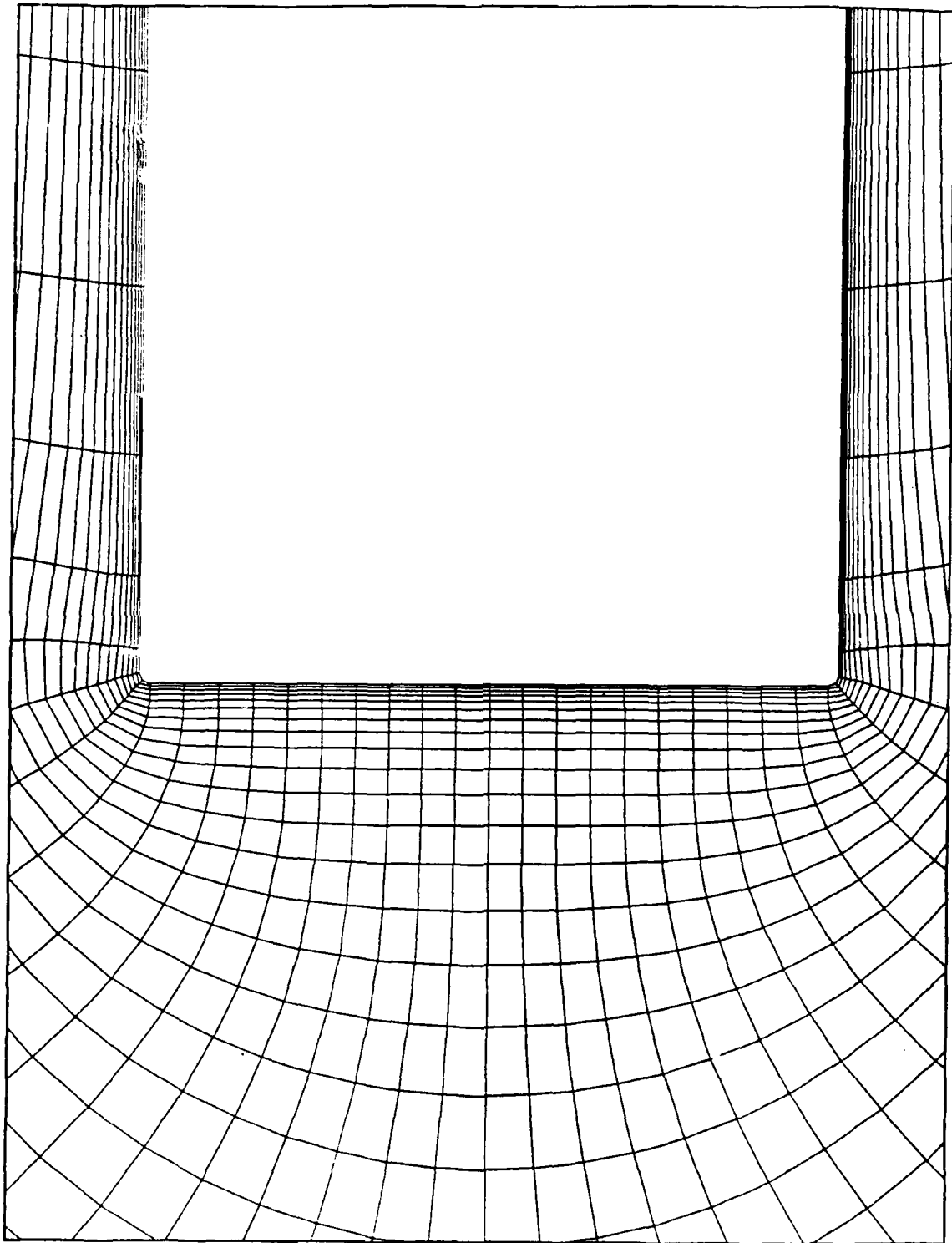
Figure 53. Wing Tip Detail of the Rectangular Section - front view
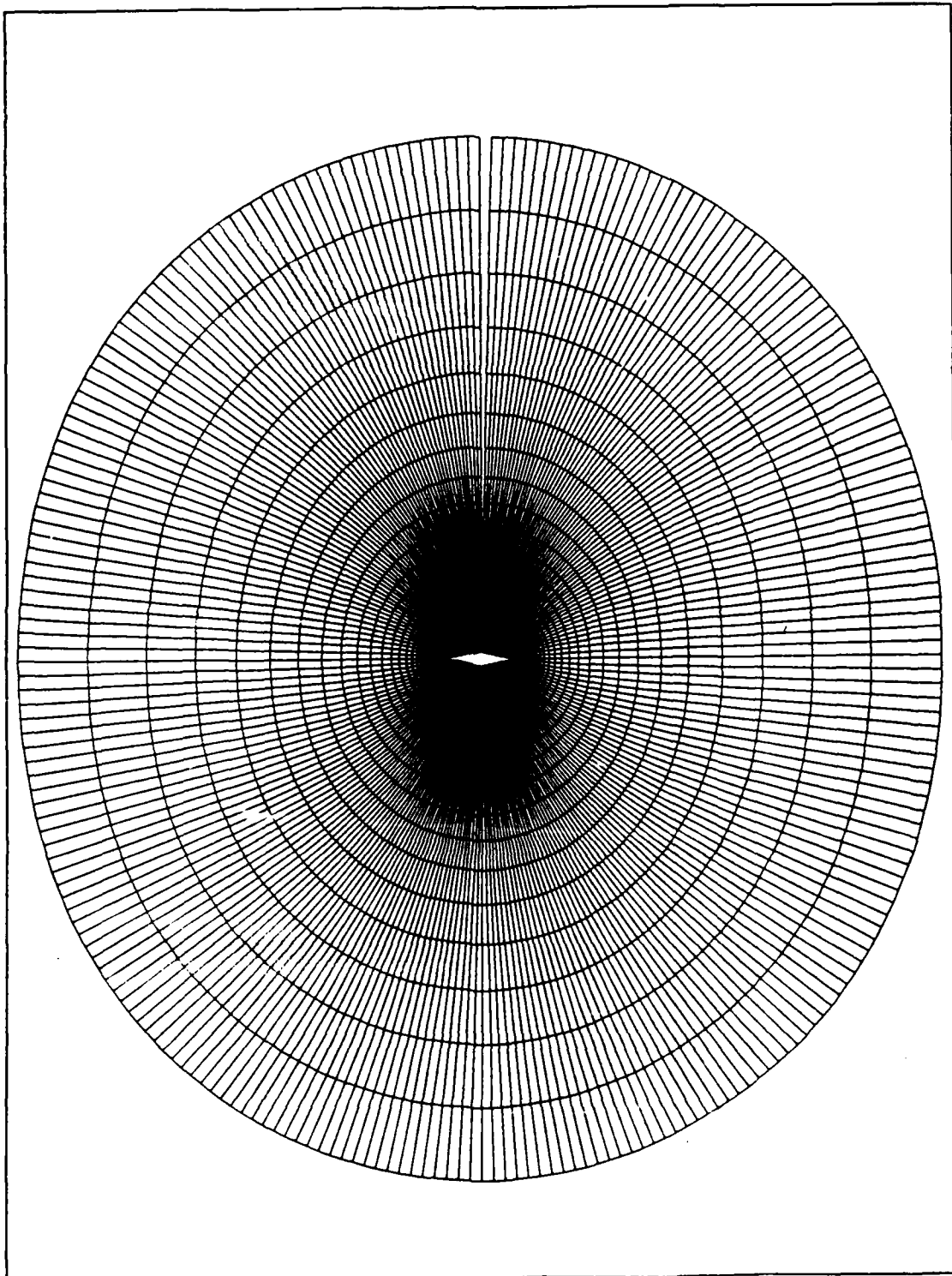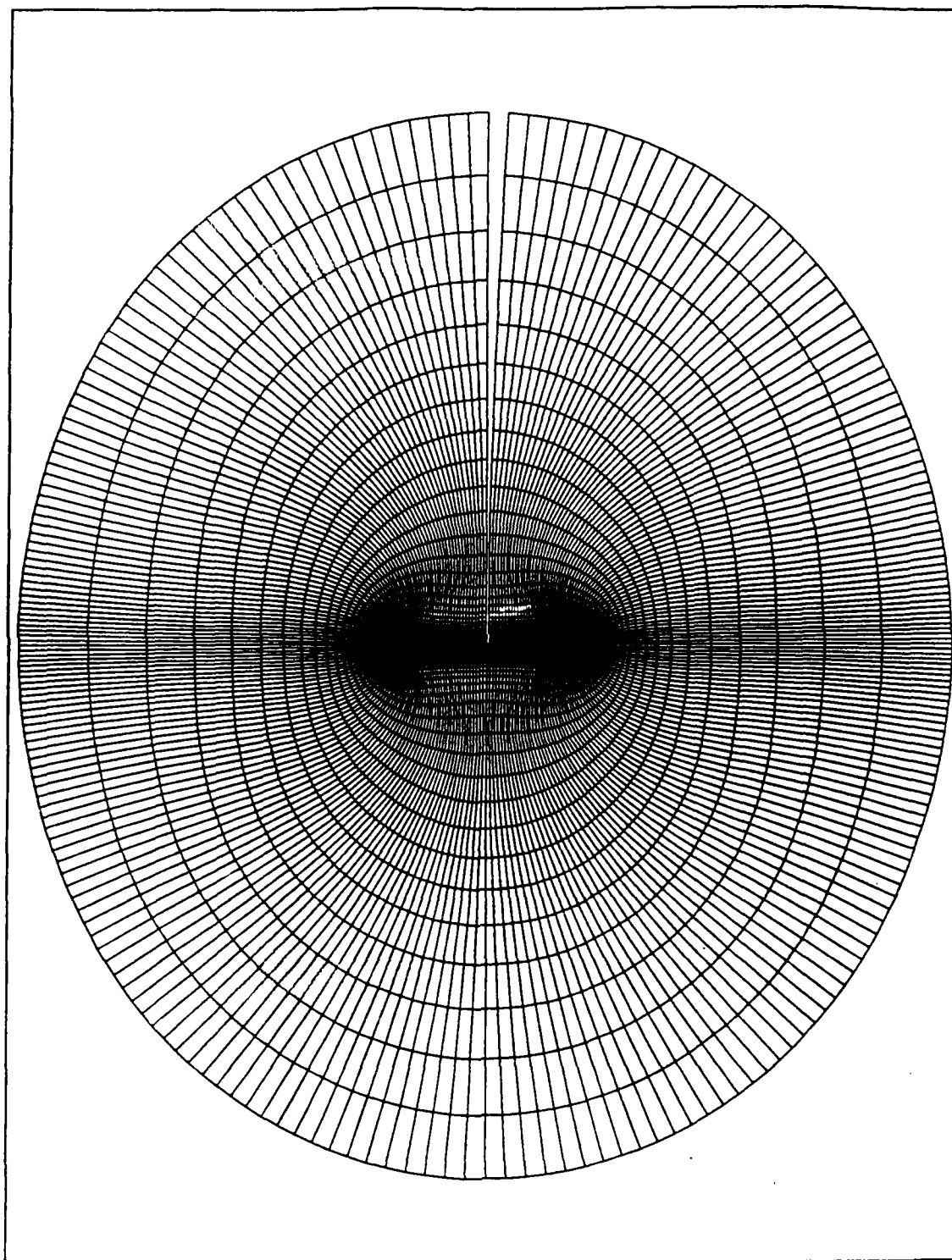
Figure 54. Typical Cross-section of the Wake - front view
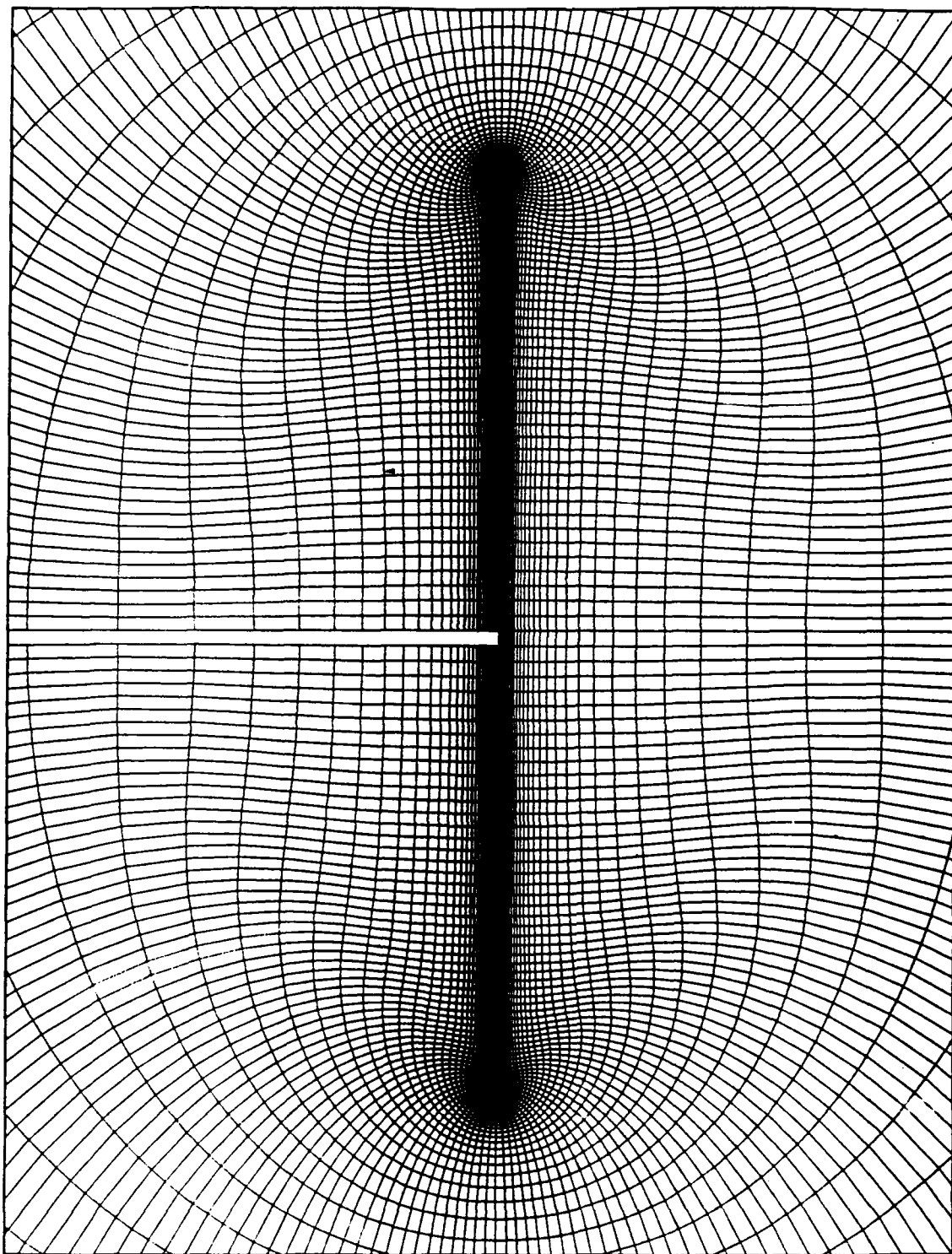
Figure 55. Near Field Grid of the Wake - front view

103
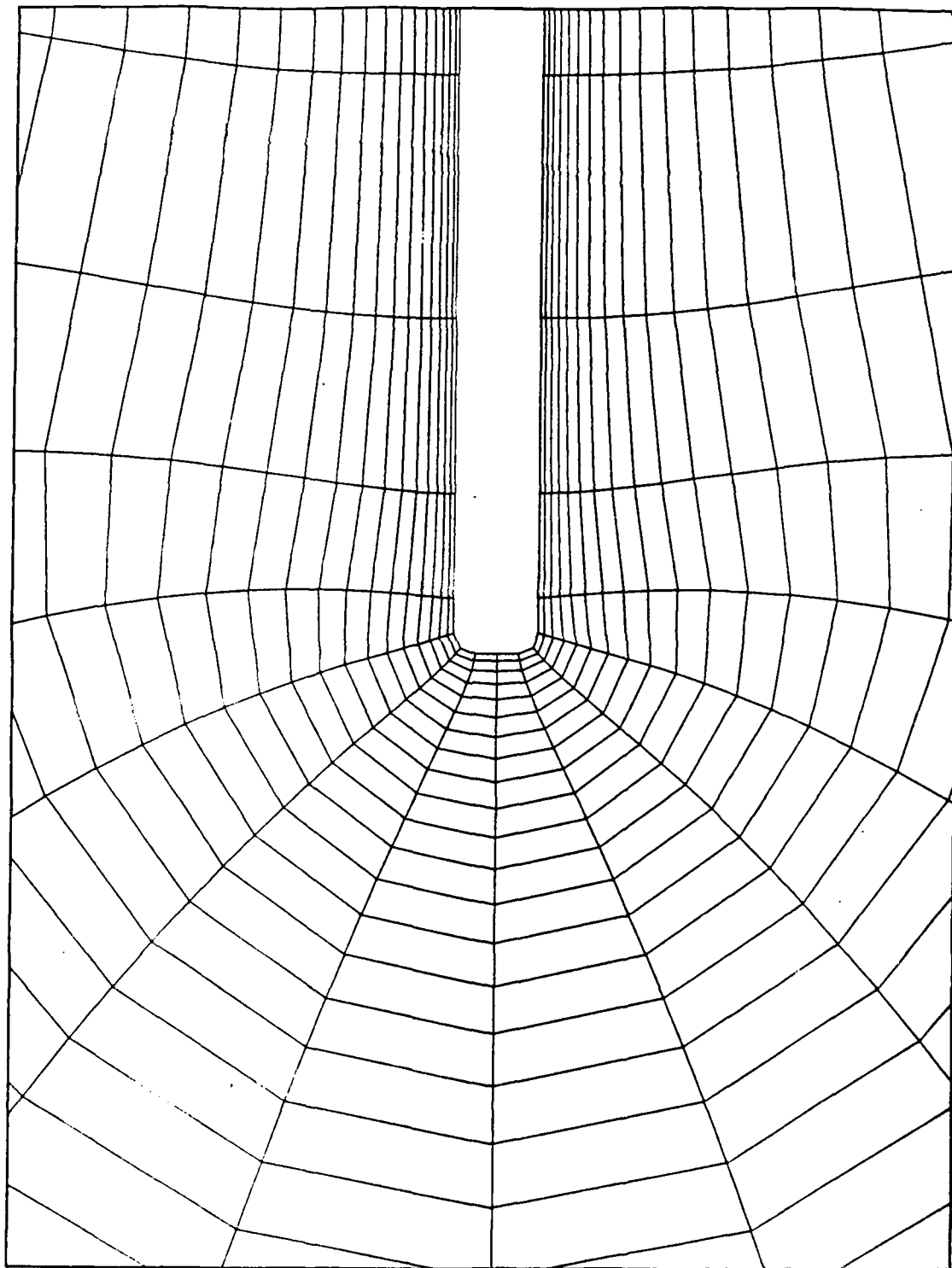
Figure 56. Edge Detail of the Wake - front view

APPENDIX D.   RESULTS AND DISCUSSION FIGURES
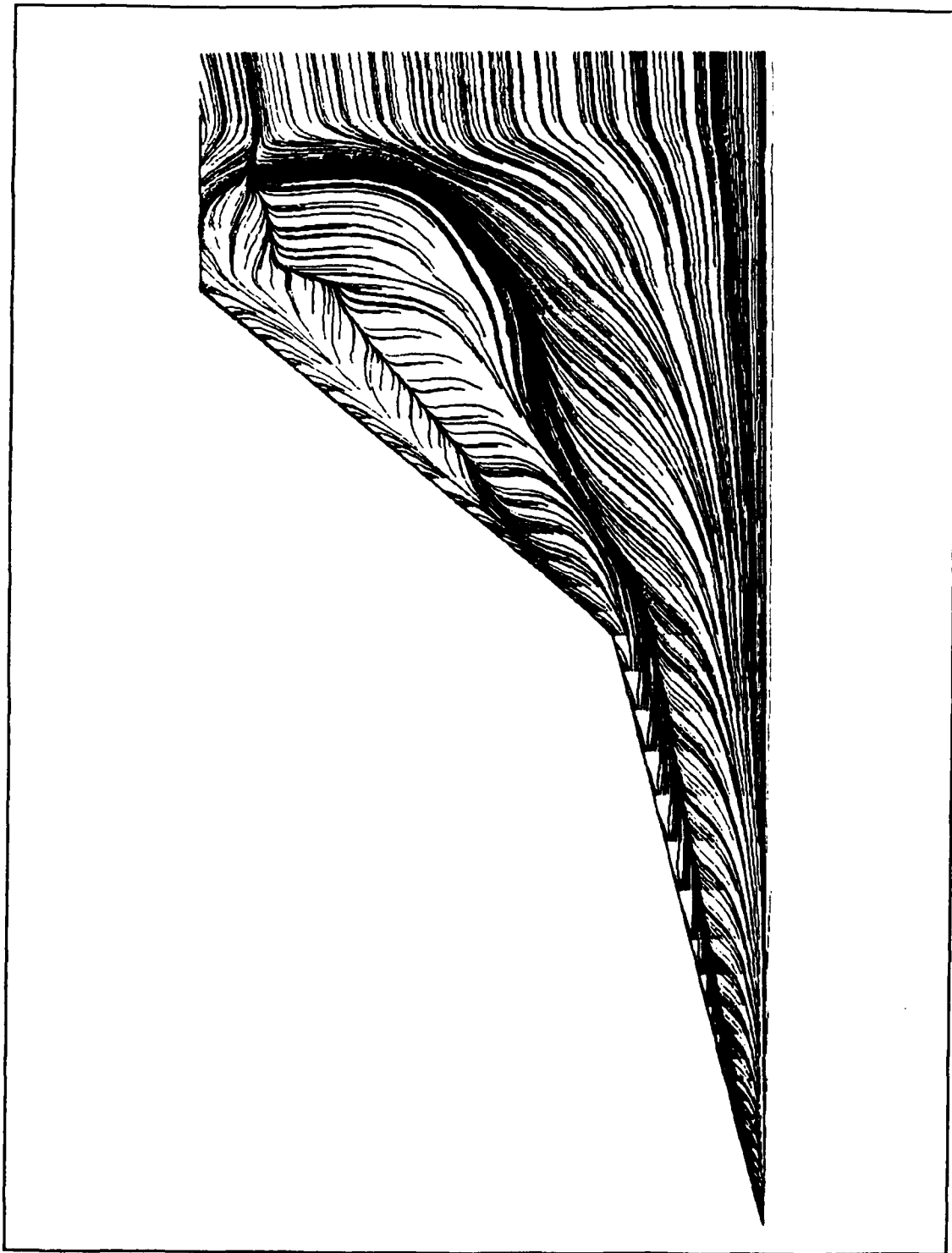
Figure 57.   Surface Flow Pattern at 10° - M = 0.22, Re = 3.8x10E6, (70x63x68)
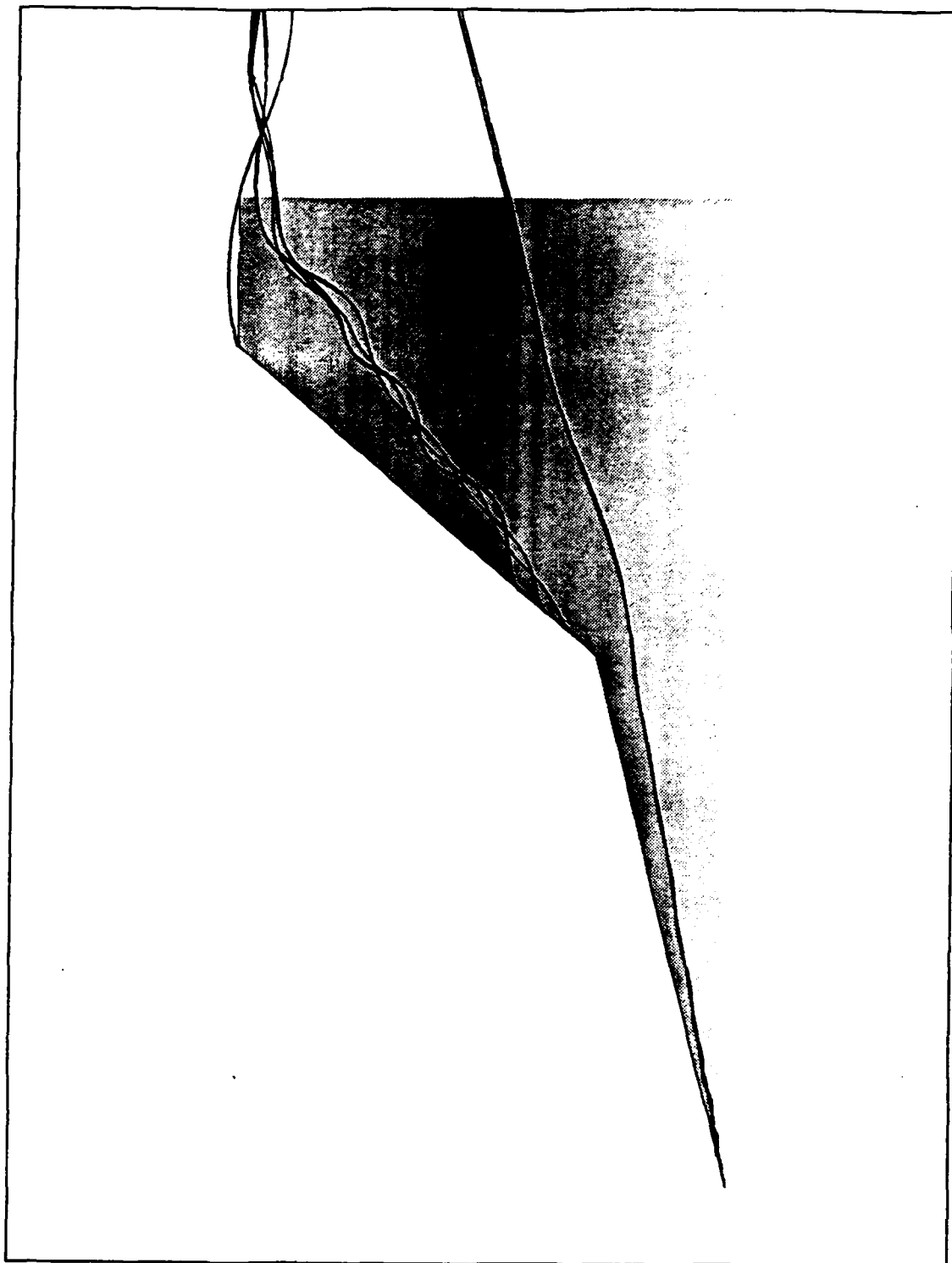
Figure 58. Particle Traces at 10° - M = 0.22, Re = 3.8x10E6, (70x63x68)
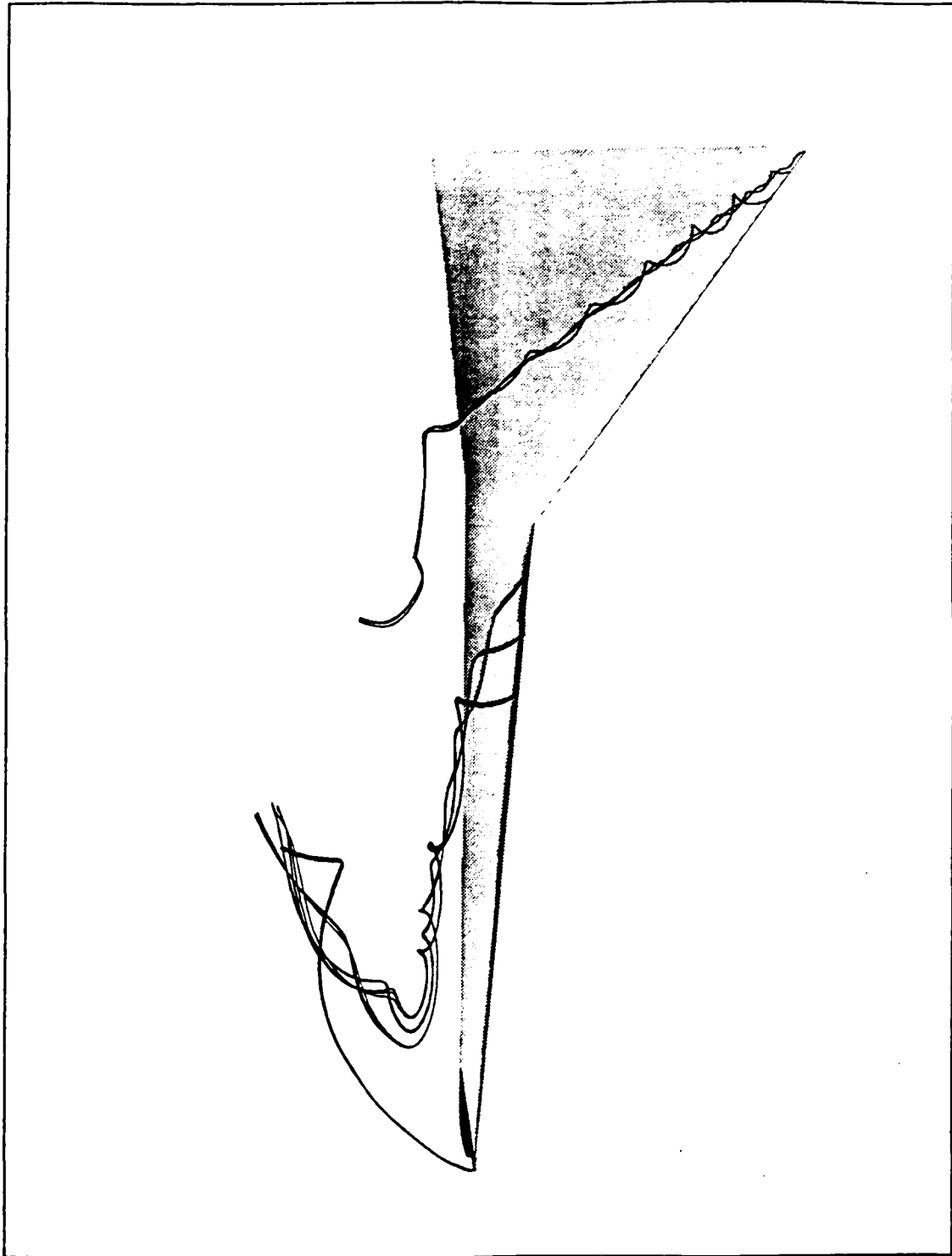
Figure 59.   Vortex Location at 10° - M = 0.22, Re = 3.8x10E6, (70x63x68)
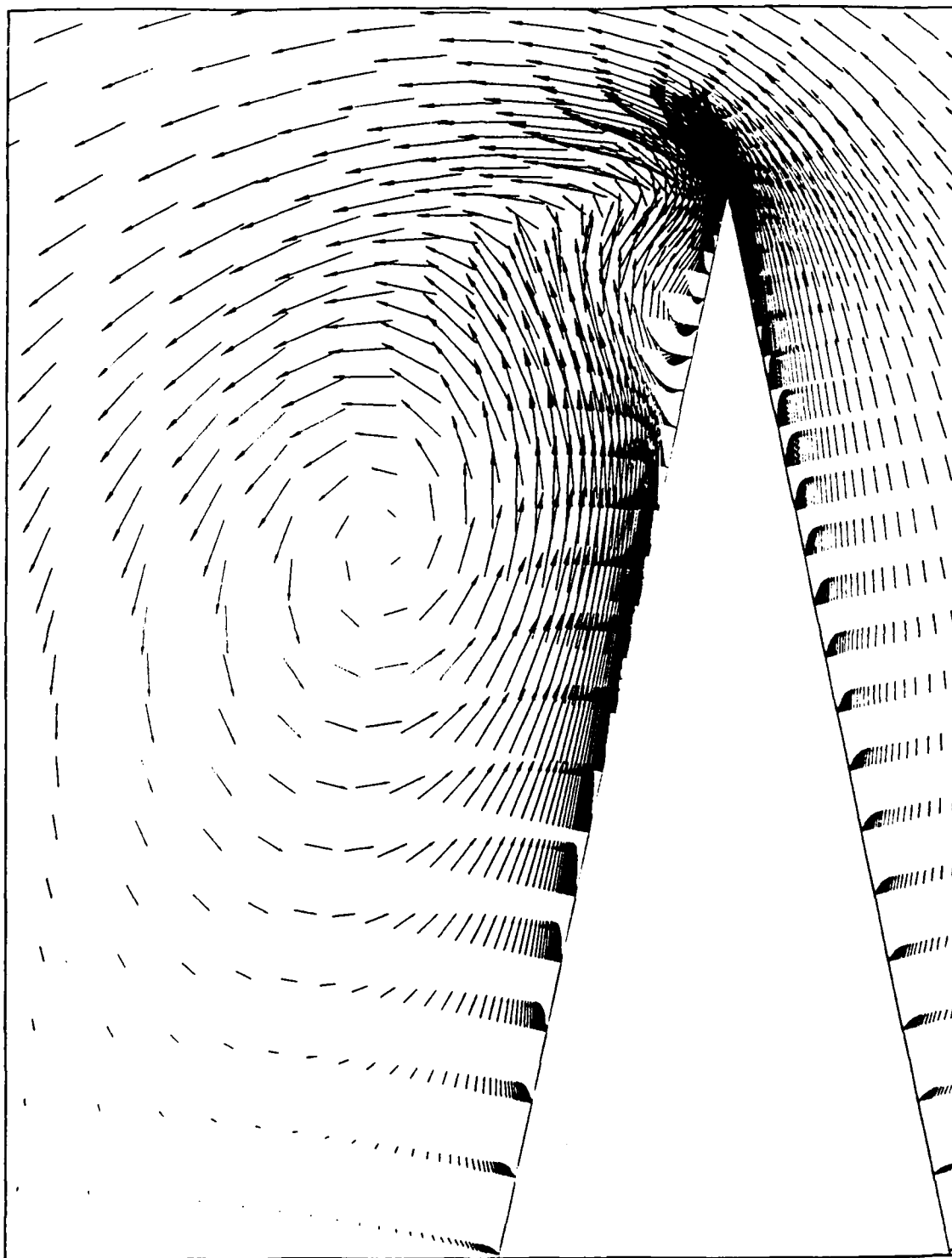
Figure 60.   Strake Velocity Vectors at 10° - M = 0.22, Re = 3.8x10E6

109
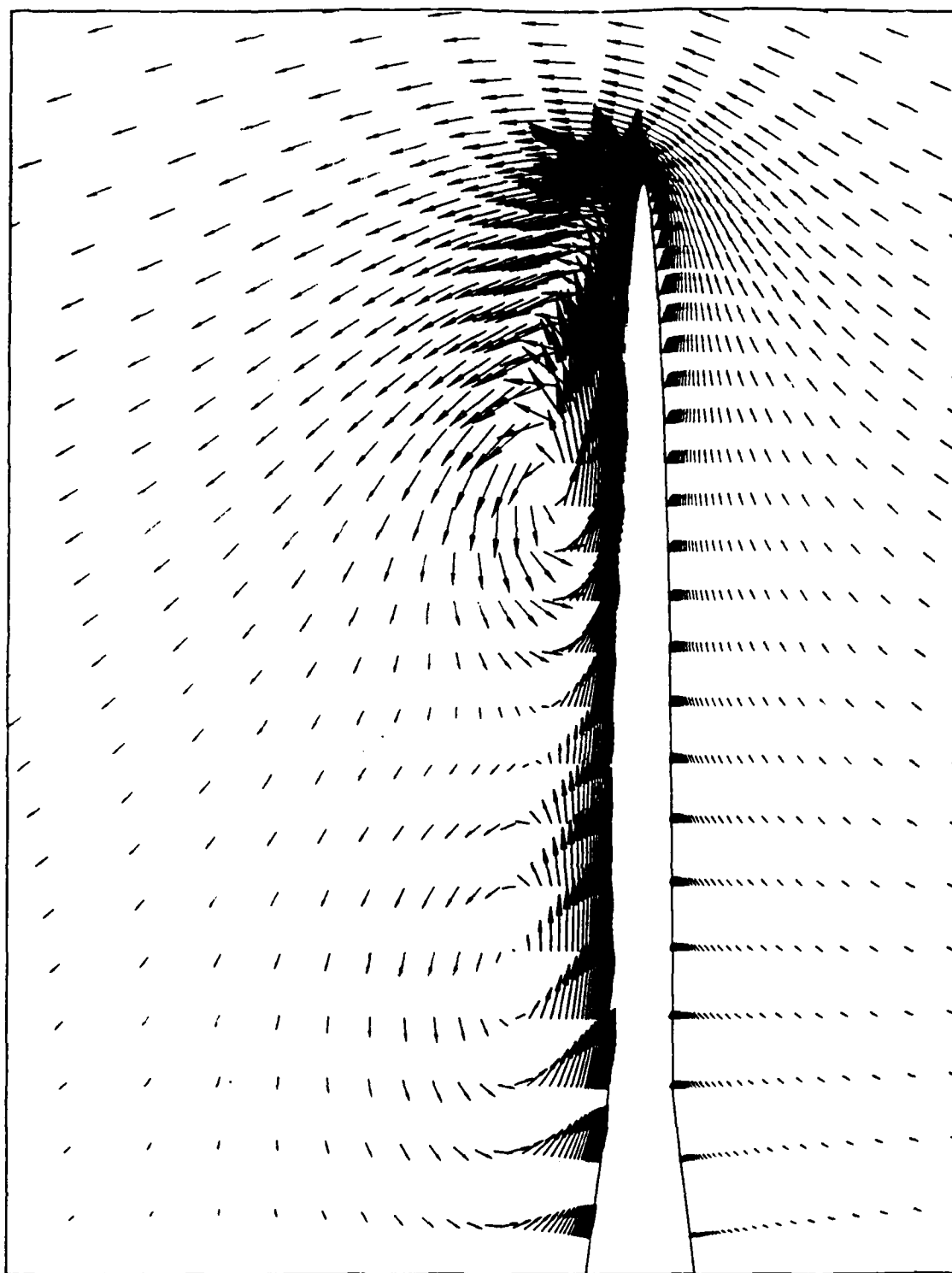
Figure 61.   Wing Velocity Vectors at 10° - M = 0.22, Re = 3.8x10E6

110
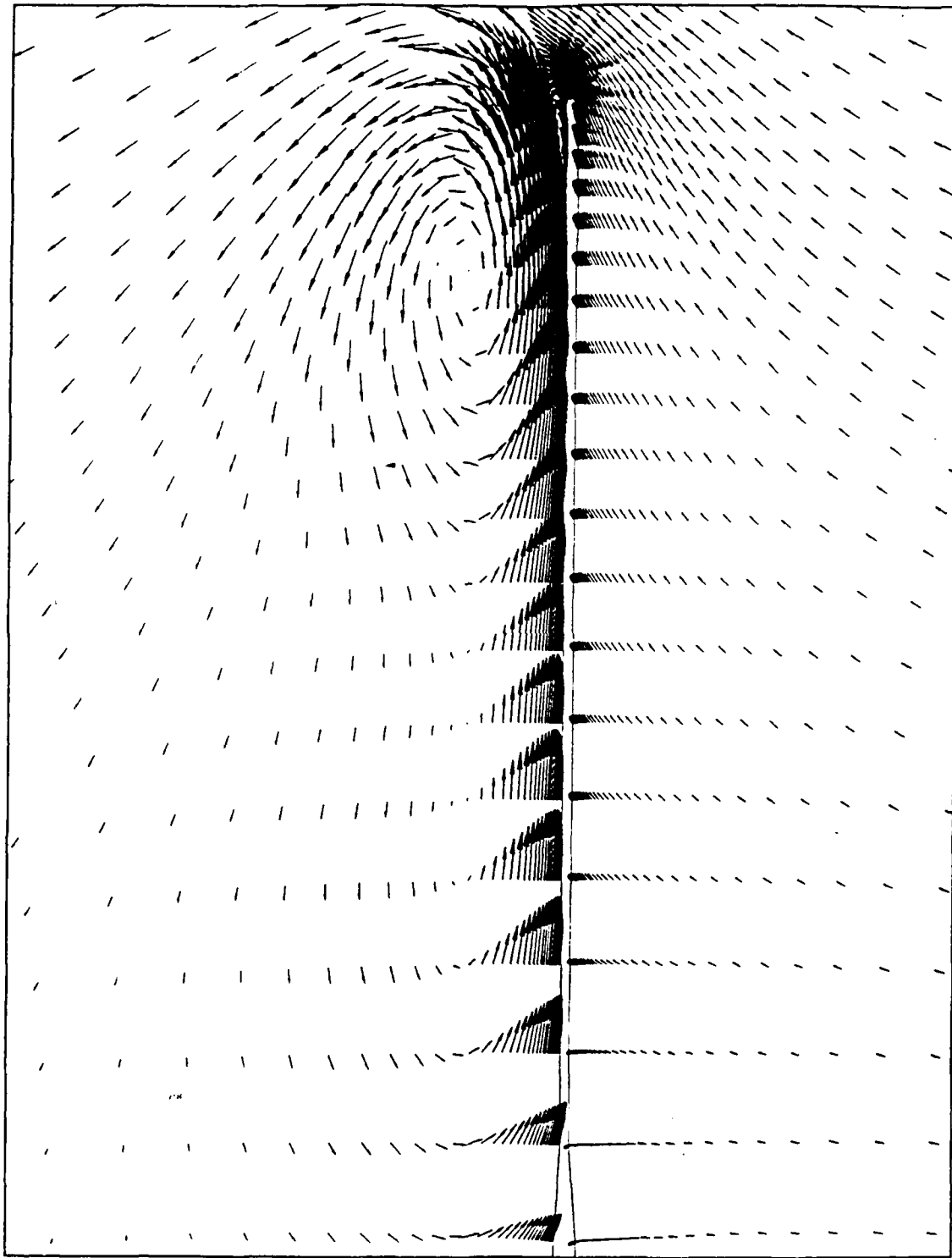
Figure 62. T. E. Velocity Vectors at 10° - M = 0.22, Re = 3.8x10E6
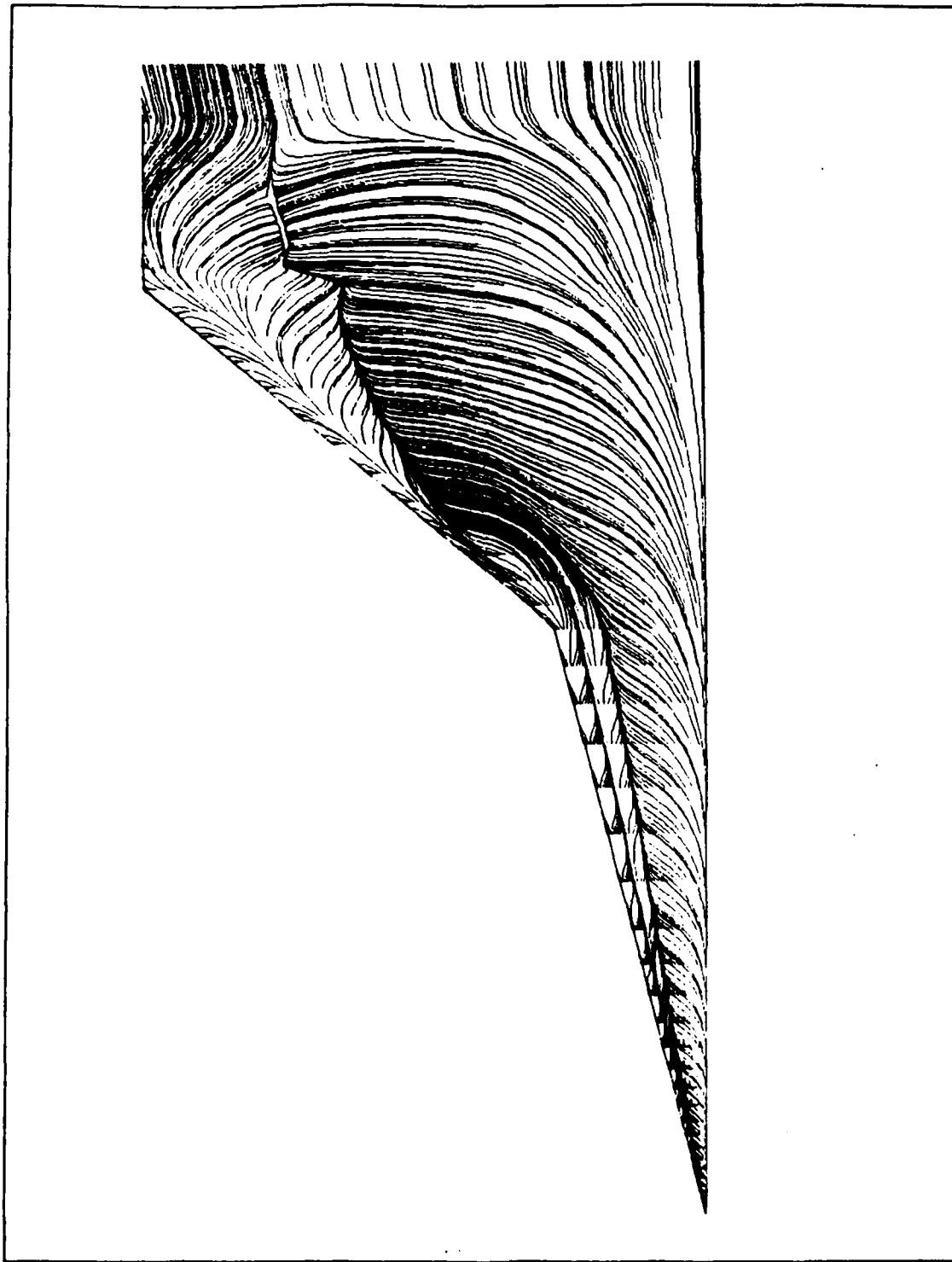
Figure 63. Surface Flow Pattern at 19°4.- M = 0.22, Re = 3.8x10E6, (70x63x68)
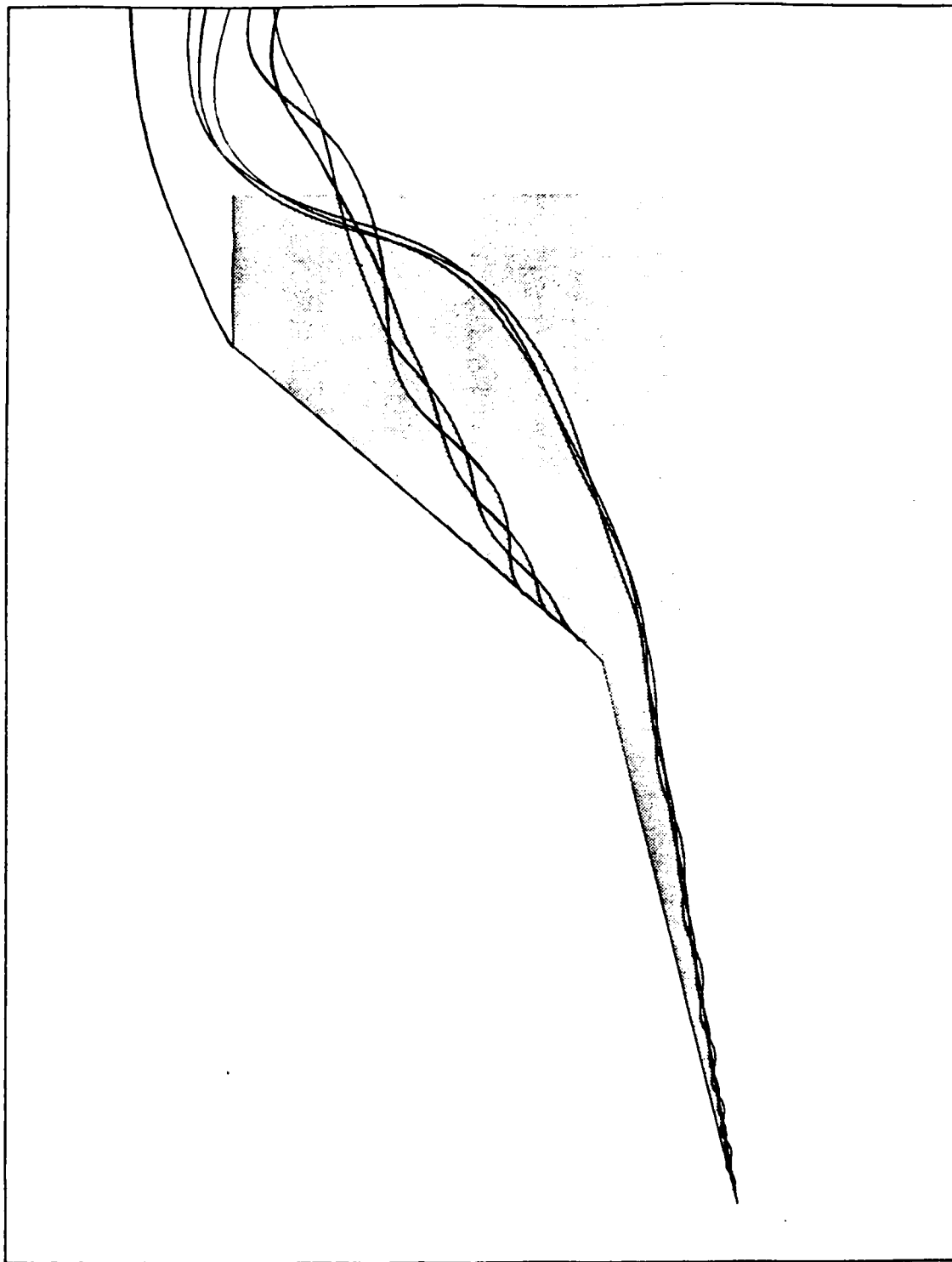
Figure 64. Particle Traces at 19° - M = 0.22, Re = 3.8x10E6, (70x63x68)

113
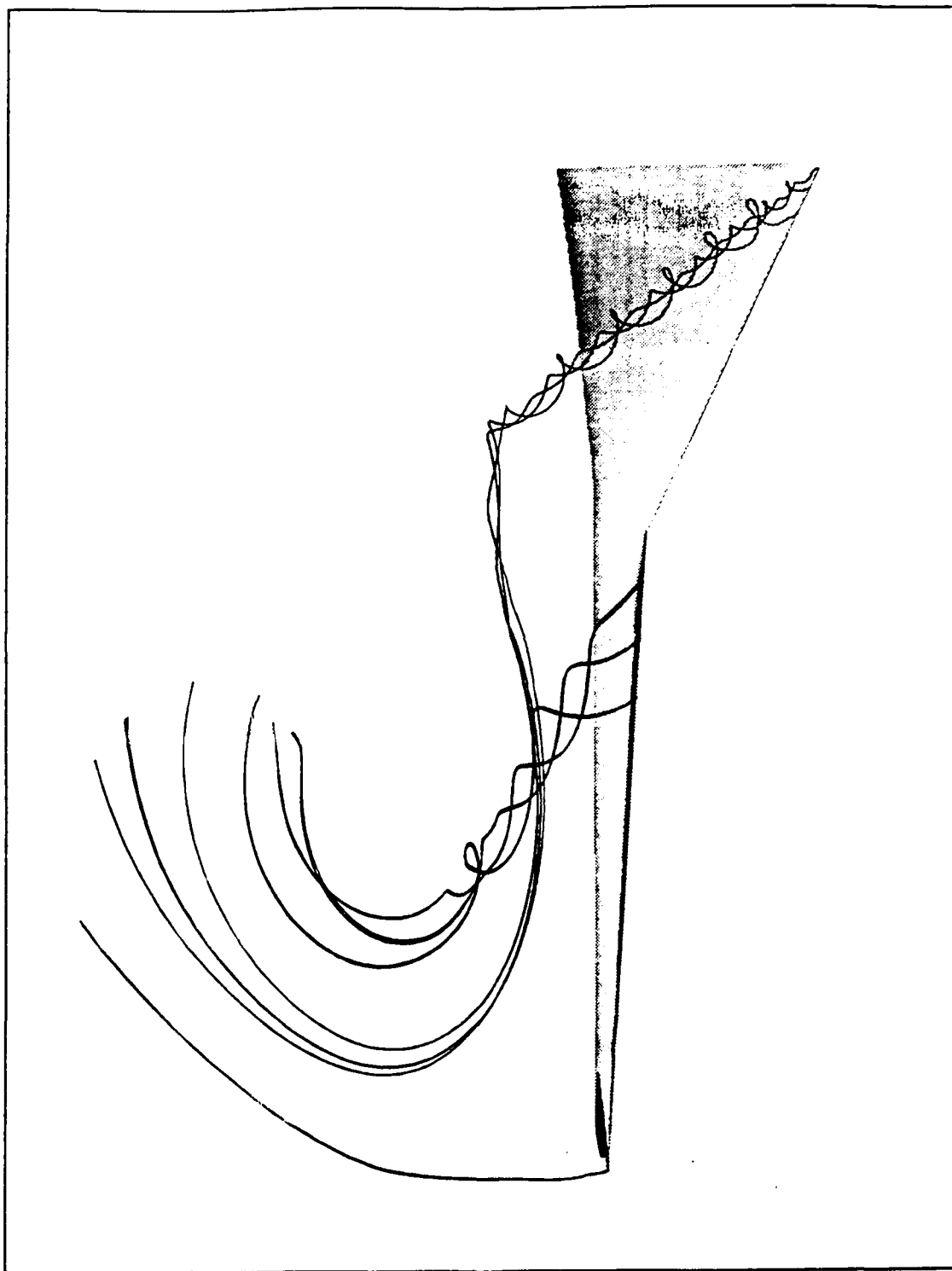
Figure 65. Vortex Location at 19°. - M = 0.22, Re = 3.8x10E6, (70x63x68)
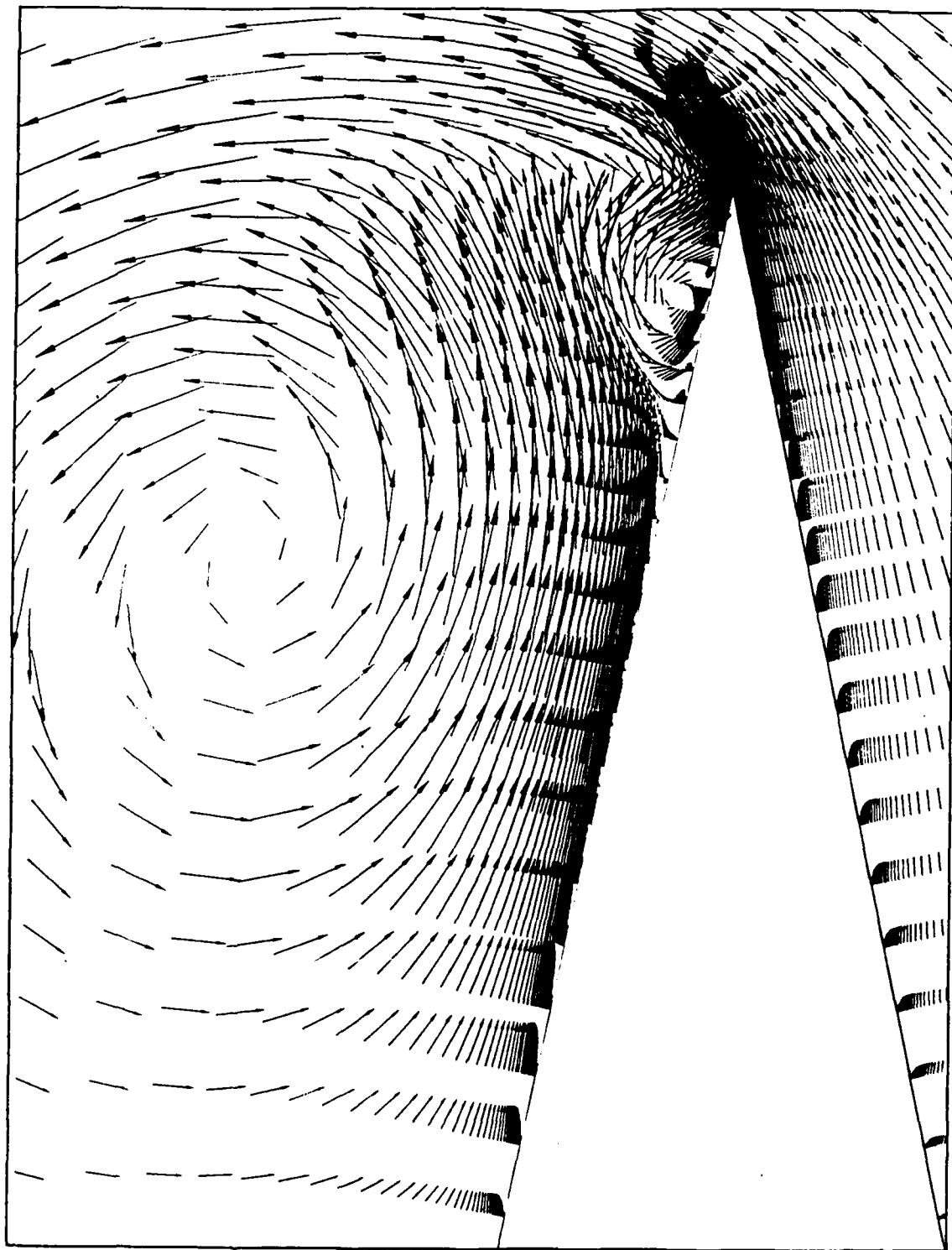
Figure 66.   Strake Velocity Vectors at 19°.- M = 0.22, Re = 3.8x10E6

115
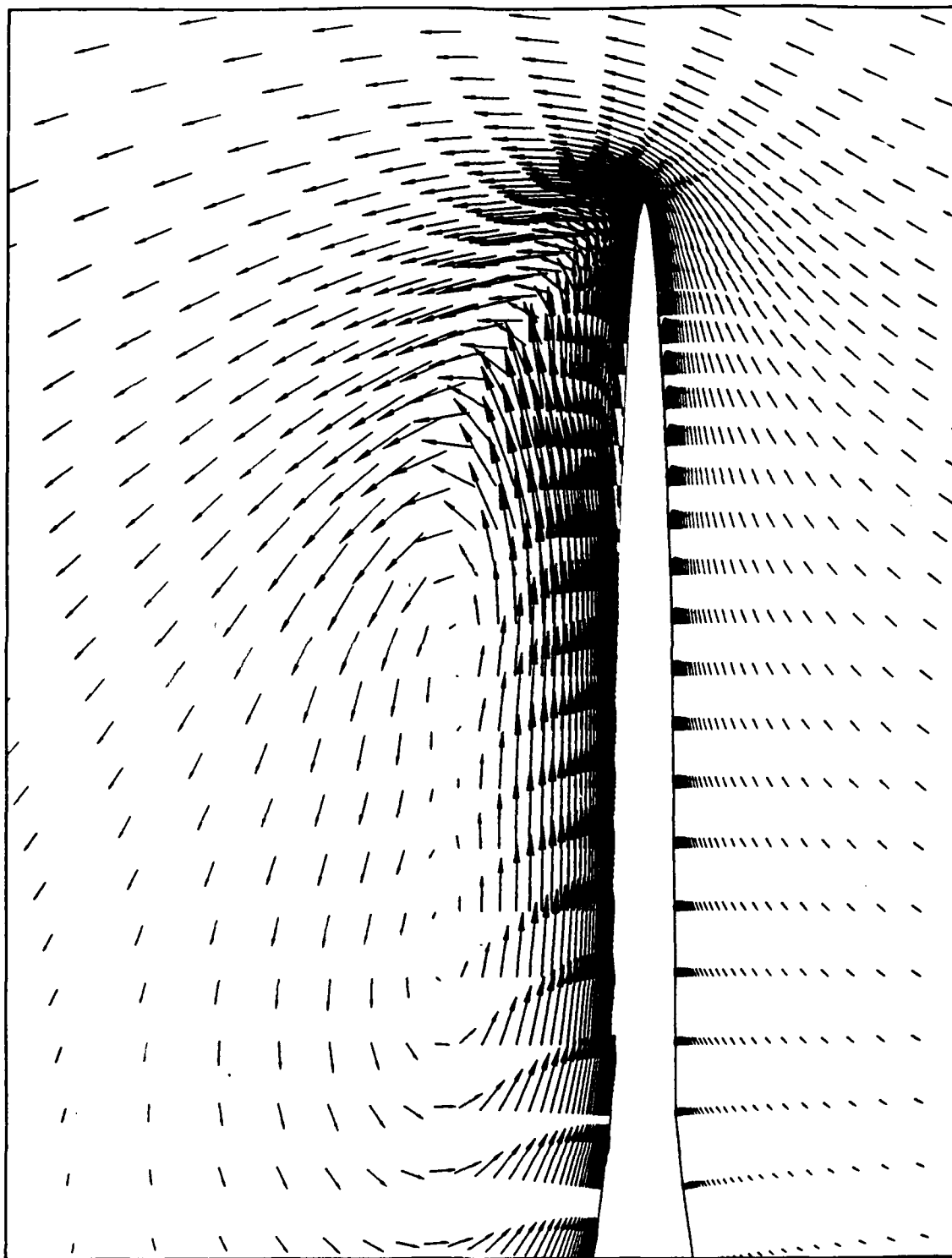
Figure 67.   Wing Velocity Vectors at 19° - M = 0.22, Re = 3.8x10E6

116
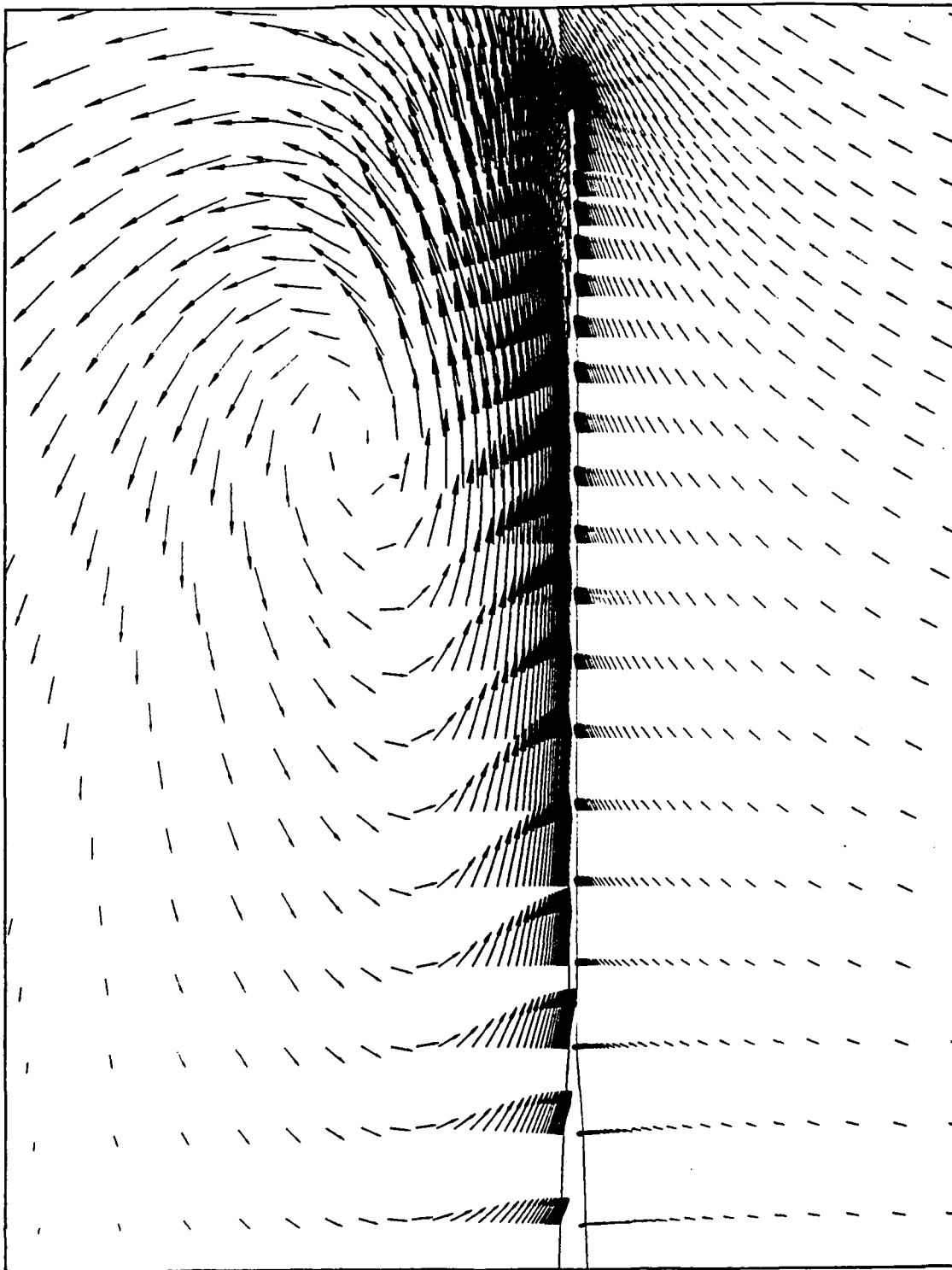
Figure 68. T. E. Velocity Vectors at 19° - M = 0.22, Re = 3.8x10E6
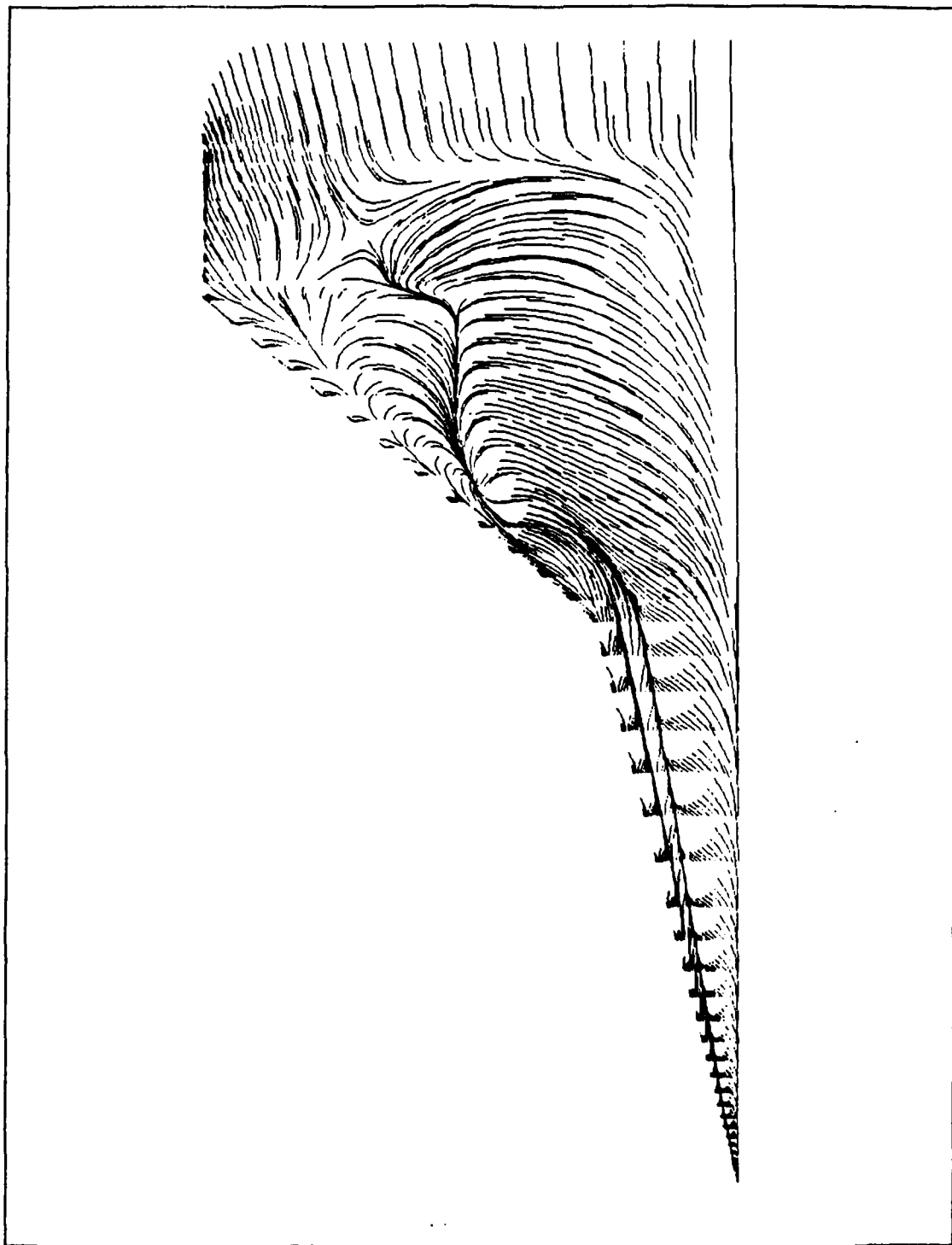
Figure 69. Surface Flow Pattern at 22.4° .- M = 0.22, Re = 3.8x10E6, (70x63x68)

118
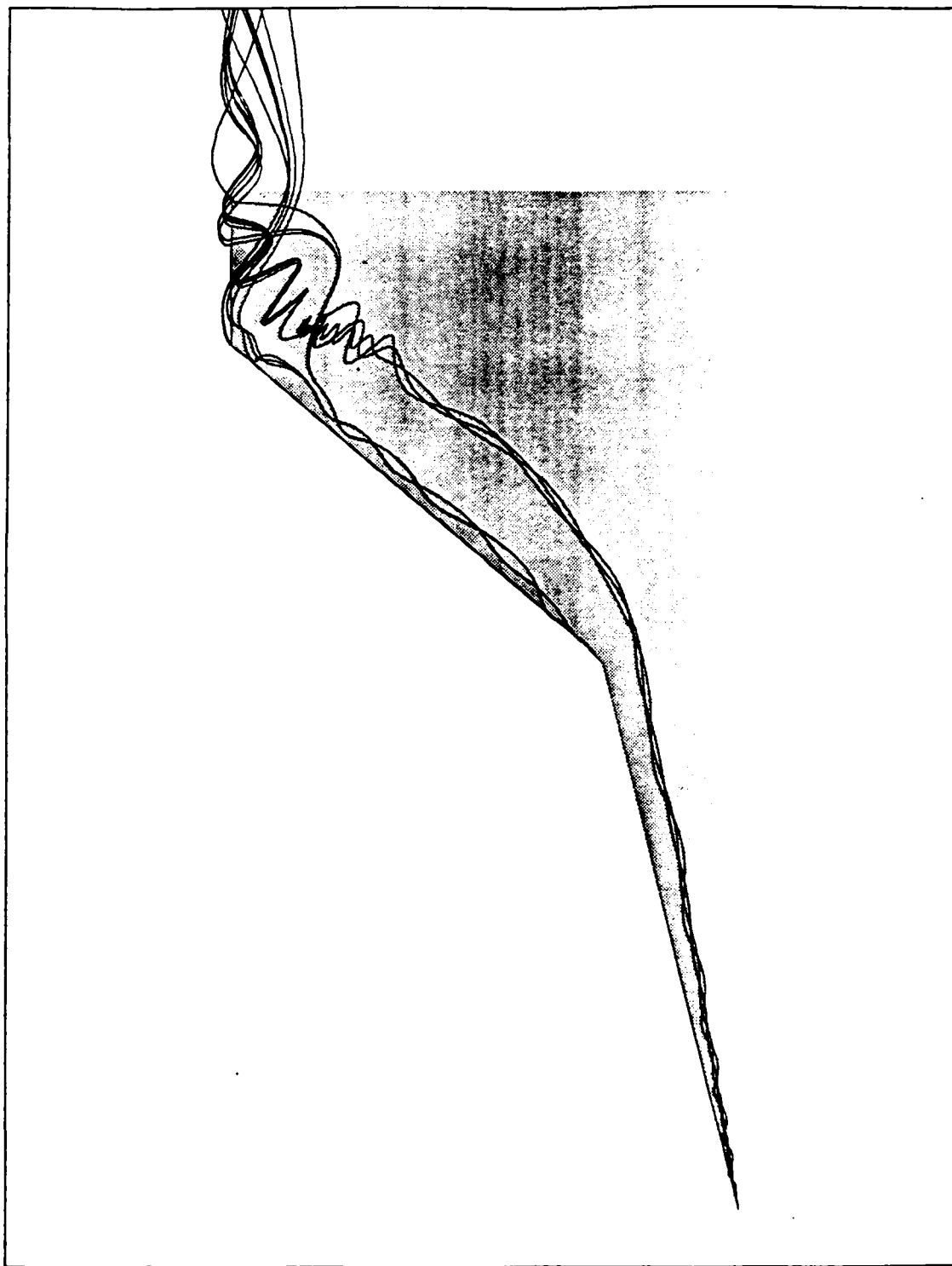
Figure 70. Particle Traces at 22.4° - M = 0.22, Re = 3.8x10E6, (70x63x68)
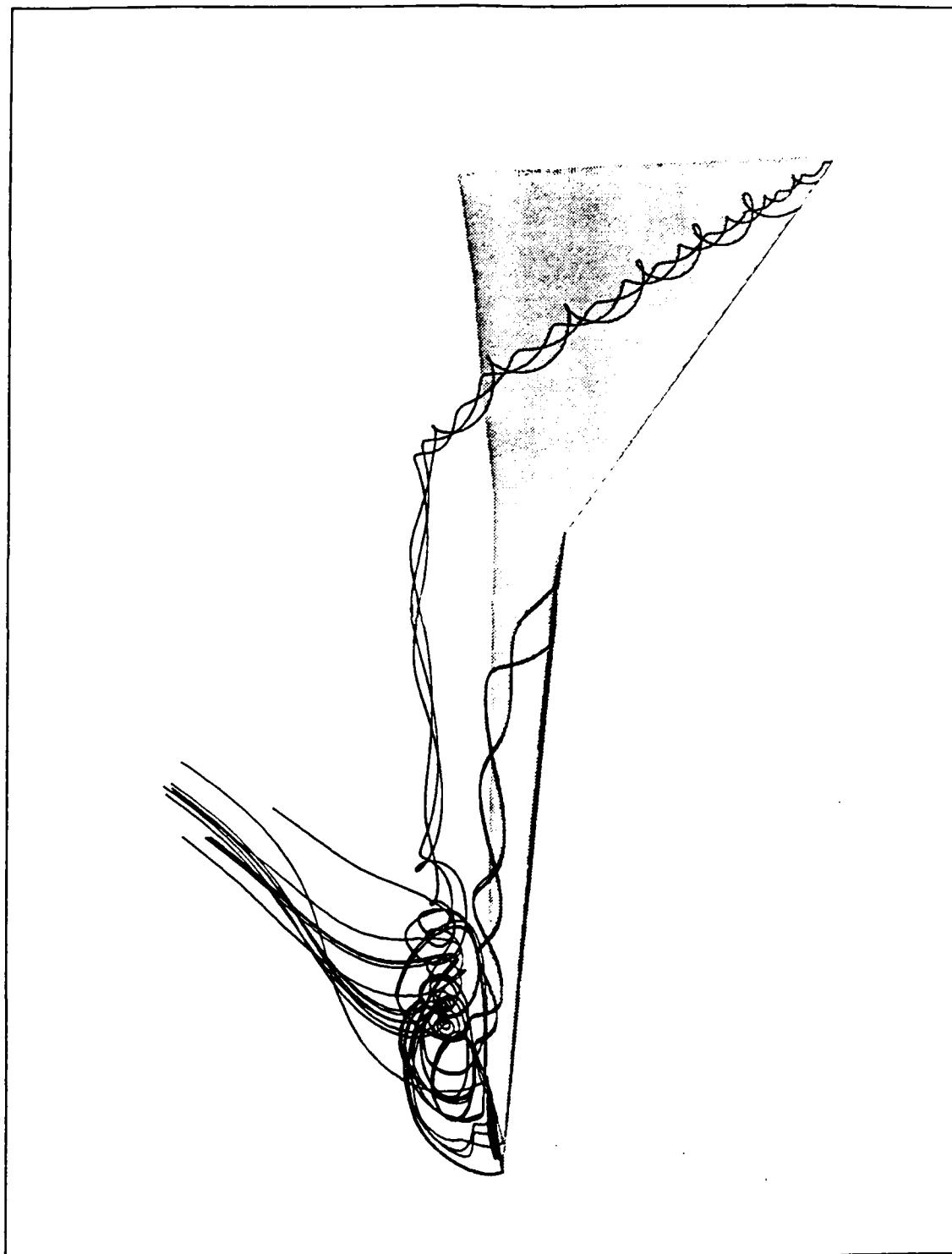
Figure 71.   Vortex Location at 22.4° - M = 0.22, Re = 3.8x10E6, (70x63x68)

120
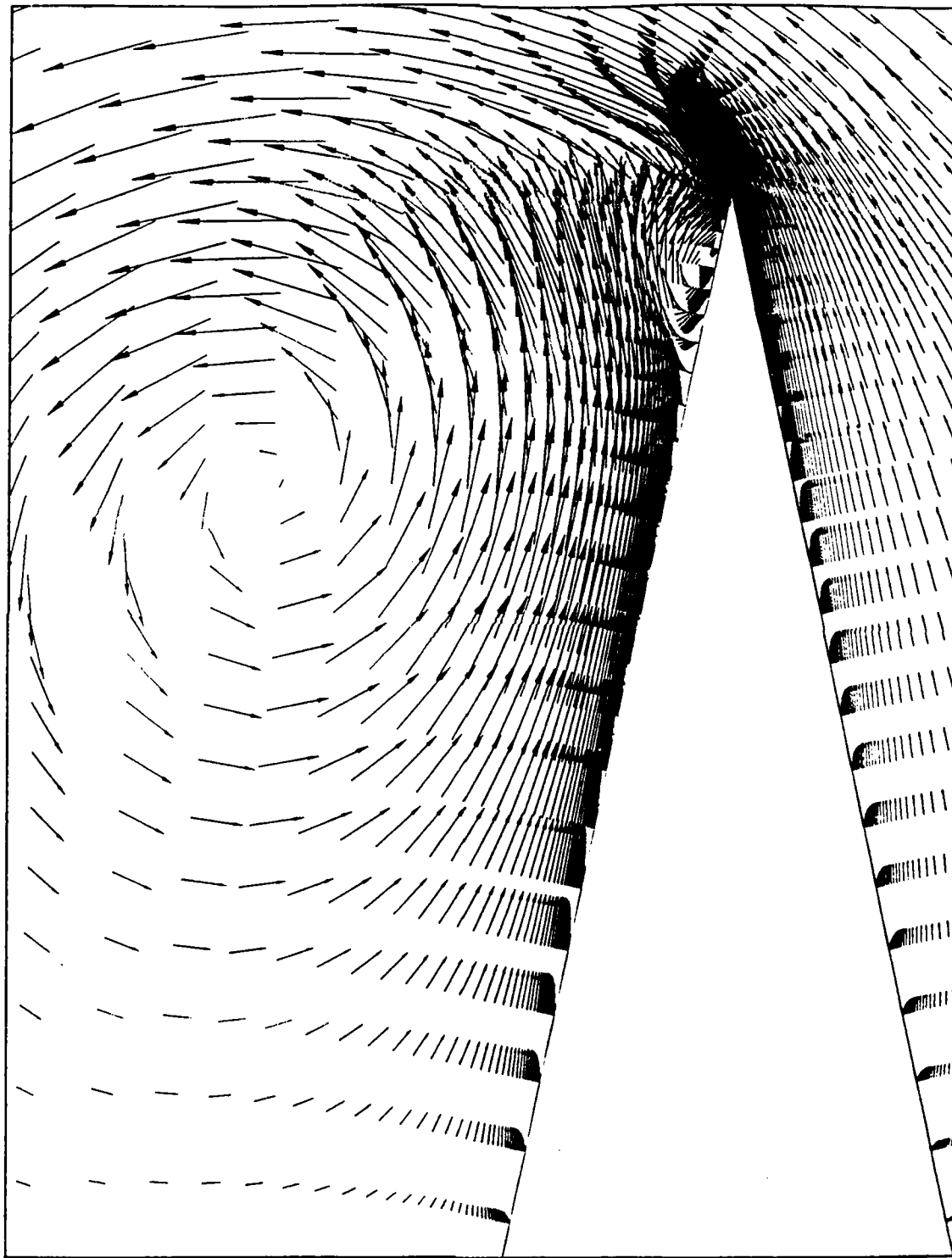
Figure 72. Strake Velocity Vectors at 22.4° - M = 0.22, Re = 3.8x10E6

121
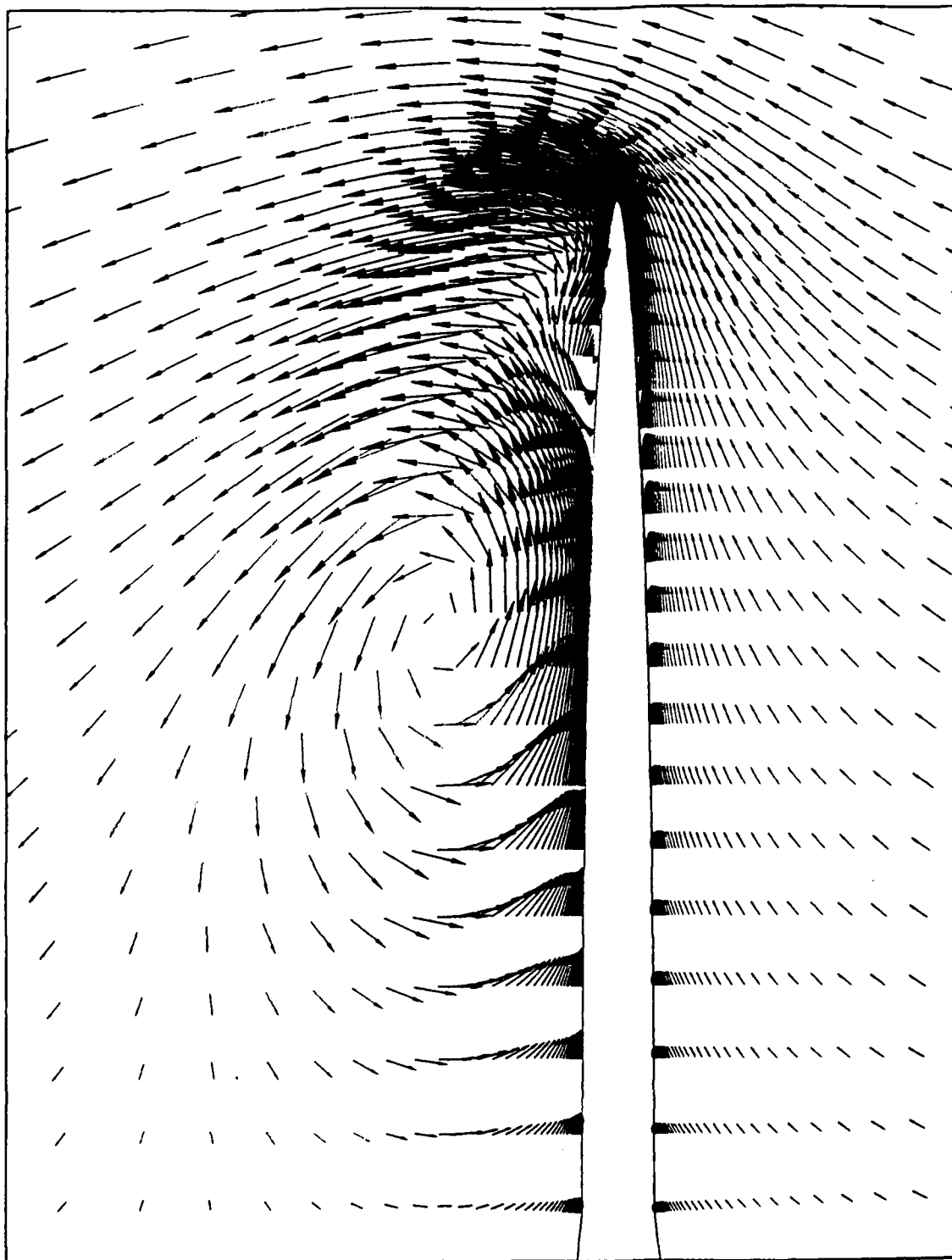
Figure 73. Wing Velocity Vectors at 22.4° - M = 0.22, Re = 3.8x10E6

122
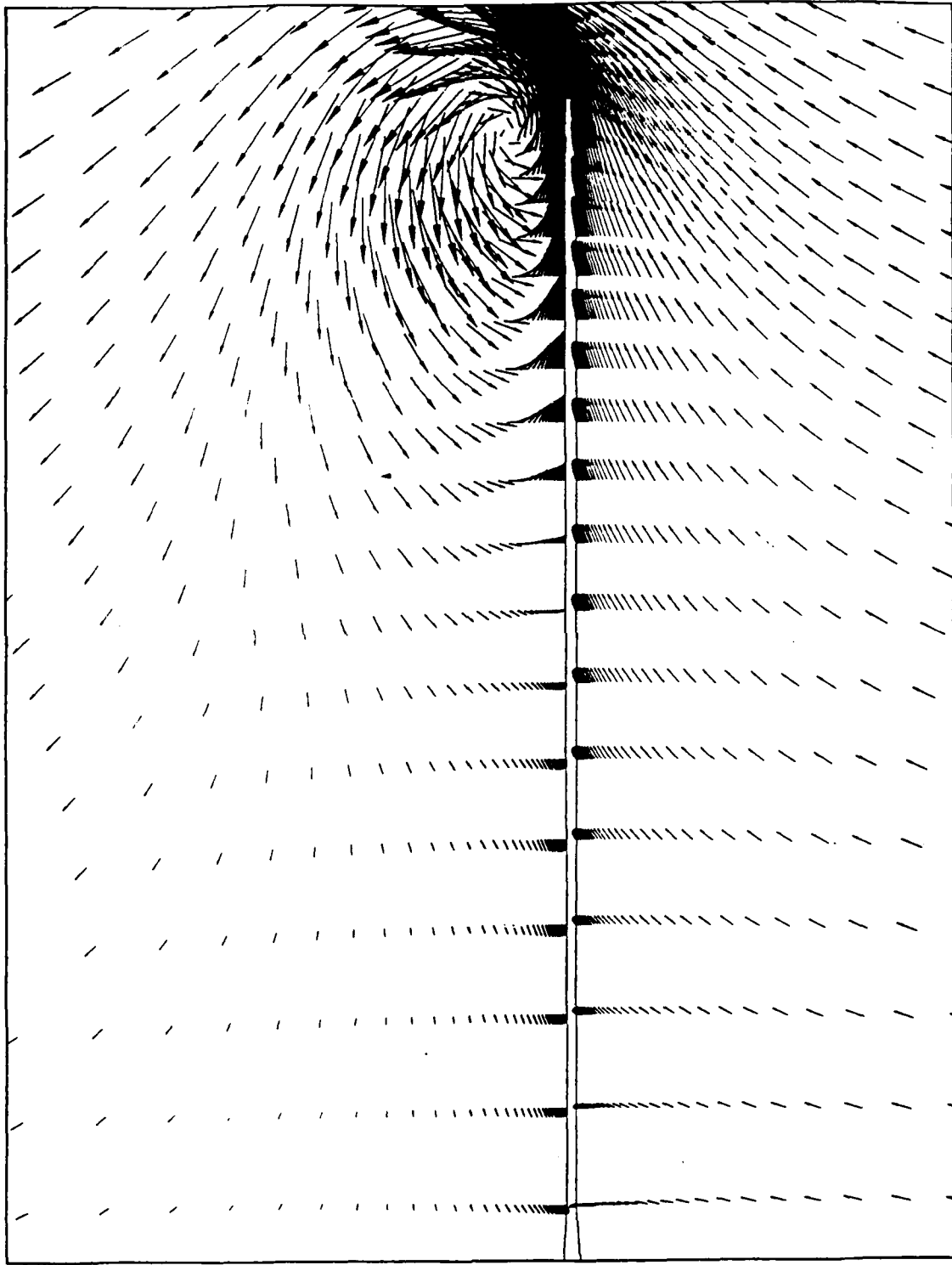
Figure 74.   T. E. Velocity Vectors at 22.4° - M = 0.22, Re = 3.8x10E6

Figure 75. Surface Pressure Coefficient at x/c = 0.40

Figure 76.   Surface Pressure Coefficient at x/c = 0.66

Figure 77.   Surface Pressure Coefficient at x/c = 0.98

APPENDIX E.   SOURCE CODE FOR SURFACE GRIDS

```
C
C    THIS IS A PROGRAM TO GENERATE A SURFACE GRID FOR A DOUBLE DELTA
C    WING AIRFOIL THAT WILL BE ANALYZED FOR MY MASTERS THESIS.
C
C    CERTAIN DATA WILL BE REQUIRED VIA A DATA FILE AND INCLUDES THE
C    FOLLOWING VARIABLES:
C
C    NOSERAD - THE RADIUS DECSCRIBING THE NOSE
C    NOSGDX1 - THE NUMBER OF GRID POINTS, IN THE X DIRECTION DESIRED
C              IN THE ROUNDED NOSE
C    NOSGDX2 - THE NUMBER OF GRID POINTS, IN THE X DIRECTION DESIRED
C              IN THE FIRST DELTA WING.  DO NOT INCLUDE LAST GRID IN
C              THE NOSE ROUNDING
C    NOSGDY  - THE NUMBER OF GRID POINTS, IN THE Y DIRECTION DESIRED
C              IN THE NOSE BODY
C    CRVGDY  - THE NUMBER OF GRID POINTS, IN THE Y DIRECTION DESIRED
C              IN THE ROUNDED LEADING EDGE
C    LEN1    - THE LENGTH FROM THE LEADING EDGE TO THE SECOND WING
C    LEN2    - THE HALF WIDTH OF THE FIRST DELTA WING
C    LEN3    - THE HALF THICKNESS OF THE FIRST DELTA WING
C
C    THE FOLLOWING IS A DEFINITION OF SOME OF THE VARIABLES USED IN
C    THE FIRST PART OF THE PROGRAM:
C
C    DELANG  - THE DELTA ANGLE USED TO GENERATE GRIDS IN THE Y DIRECTION
C    DISTA   - THE DISTANCE FROM THE TIP OF THE AIRFOIL TO THE LAST
C              GRID GENERATED IN THE NOSE ROUNDING
C    DELDIST - THE DELTA DISTANCE IN THE X DIRECTION USED IN ROUNDING
C              THE NOSE
C    ANG     - THE ANGLE USED TO CALCULATE THE SPECIFIC GRIDS IN THE
C              Y DIRECTION

C    RAD     - THE RADIAL DISTANCE USED TO CALCULATE GRIDS IN THE Y
C              DIRECTION
C    X1      - THE DISTANCE FROM THE CENTER OF THE NOSE TO THE TRAILING
C              EDGE OF THE FIRST DELTA WING
C    ANG1    - THE ANGLE BETWEEN THE CENTERLINE AND TRAILING EDGE
C    ANG2    - THE ANGLE BETWEEN THE TRAILING EDGE AND LEADING EDGE
C              OF THE FIRST DELTA WING
C    THETA1  - THE ANGLE FORMED BY THE NOSE ROUNDING, IT WILL BE
C              PERPENDICULAR TO THE LEADING EDGE OF THE FIRST DELTA WING
C    DISTB   - THE DISTANCE FROM THE TRAILING EDGE TO THE NOSE RADIUS
C    INUM    - THE TOTAL NUMBER OF LINEAR GRIDS ON THE UPPER AND LOWER
C              SURFACE
C    DISTF   - THE DISTANCE IN THE Z DIRECTION TRAVERSED BY THE THICKNESS
C              OF THE LEADING EDGE
C    THETA2  - THE ANGLE FORMED BY THE THICKNESS OF THE FIRST DELTA WING
C    DISTC   - THE HALF WIDTH IN THE Y DIRECTION USED TO GENERATE Y AND
C              Z DIRECTION GRIDS
C    DISTD   - THE HALF THICKNESS USED IN SAME CALCULATIONS AS DISTC
C    THETA3  - SAME TYPE OF ANGLE AS THETA1 BUT USED FOR Y AND Z GRIDS
C    DISTE   - THE DISTANCE IN THE Y DIRECTION FOR Y AND Z GRID
C    DELY    - THE DELTA Y DISTANCE FOR THE DISTE VARIABLE
C
      DIMENSION X(170,150,1),Y(170,150,1),Z(170,150,1)
      INTEGER NOSGDX1,NOSGDX2,NOSGDY,CRVGDY,BODGDX1,BODGDX2,NOSGDX,
     *        SIDGRD,WAKGRD,CRVGDY1
      REAL LEN1,LEN2,LEN3,LEN4,LEN5,LEN6,NOSERAD,NOSERAD1
```

```
        OPEN(UNIT=10,FILE='ddwsg.in',STATUS='OLD')
C       OPEN(UNIT=12,FILE='ddwsg.dat',STATUS='NEW')

        READ(10,*)  NOSERAD,NOSGDX1,NOSGDX4,NOSGDX5,NOSGDY
        READ(10,*)  CRVGDY,SIDGRD,WAKGRD
        READ(10,*)  BODGDX1,BODGDX2,THETA4
        READ(10,*)  LEN1,LEN2,LEN3
        READ(10,*)  LEN4,LEN5,LEN6
        READ(10,*)  DISTX1A,DISTX1B,DISTX2,DISTX3,DISTX4
        READ(10,*)  C1A,E1A,C1B,E1B,C2A,E2A,C2B,E2B
        READ(10,*)  C3A,E3A,C3B,E3B
        READ(10,*)  WAKLEN

        PRINT *,'YES1'

C
C  THIS SECTION DOES SOME PRELIMINARY CALCULATIONS ON THE WING
C

        PI=2.0*ASIN(1.0)
        DELANG=PI/(NOSGDY-1)
        X1=((LEN1-NOSERAD)**2.0+LEN2**2.0)**0.5
        DUMMY=LEN2/X1
        ANG1=ASIN(DUMMY)
        DUMMY=NOSERAD/X1
        ANG2=ACOS(DUMMY)
        THETA1=PI-ANG1-ANG2

C
C  THIS SECTION DOES THE ROUNDING OF THE NOSE TO AVOID SINGULARITIES
C

        DO 10 IX=1,NOSGDX1

          DO 10 IY=1,NOSGDY

            IF(IX.EQ.1)THEN

                X(IX,IY,1)=0.0
                Y(IX,IY,1)=0.0
                Z(IX,IY,1)=0.0

            ELSE

                X(IX,IY,1)=NOSERAD-COS((THETA1/(NOSGDX1-1))*(IX-1))
     *                    *NOSERAD
                ANG=PI/2.0+(IY-1)*DELANG
                RAD=NOSERAD*SIN((THETA1/(NOSGDX1-1))*(IX-1))
                Y(IX,IY,1)=-1.0*COS(ANG)*RAD
                Z(IX,IY,1)=SIN(ANG)*RAD

            ENDIF

 10     CONTINUE

        PRINT *,'YES2'

C
C THIS SECTION DOES SOME PRELIMINARY CALCULATIONS ON THE WING
```

```
C

      DISTB=LEN1-NOSERAD+COS(THETA1)*NOSERAD
      INUM=NOSGDY-CRVGDY
      DISTF=LEN3-SIN(THETA1)*NOSERAD
      THETA2=ATAN(DISTF/DISTB)
      CUMDIST=0.0

C
C  THIS SECTION COMPUTES GRID POINTS FOR THE NOSE UP TO THE SECOND WING
C

      CRVGDY1=NOSGDY-104
      NOSGDX2=NOSGDX4/2+NOSGDX5/2

      DO 20 IX=1,NOSGDX2

         IF(IX.LT.4 ) CRVGDY1=CRVGDY1-2
         IF(IX.GE.4.AND.IX.LT.6) CRVGDY1=CRVGDY1-2
         IF(IX.GE.6.AND.IX.LT.NOSGDX2) CRVGDY1=CRVGDY1-2
         IF(CRVGDY1.LT.7) CRVGDY1=7
         INUMN=NOSGDY-CRVGDY1
         CUMDELY=0.0

         IF(IX.LE.NOSGDX4/2)THEN

            CALL STRCH4(DISTB,NOSGDX4,IX,DELDIST,DISTX1A)

         ELSE

            CALL STRCH4(DISTB,NOSGDX5,IX,DELDIST,DISTX1B)

         ENDIF

         ICOUNT=0
         CUMDIST=CUMDIST+DELDIST
         DISTC=SIN(THETA1)*NOSERAD+(CUMDIST)/TAN(THETA1)
         DISTD=SIN(THETA1)*NOSERAD+(CUMDIST)*TAN(THETA2)
         X1=((DISTC-NOSERAD)**2.0+DISTD**2.0)**0.5
         DUMMY=DISTD/X1
         ANG1=ASIN(DUMMY)
         DUMMY=NOSERAD/X1
         ANG2=ACOS(DUMMY)
         THETA3=PI-ANG1-ANG2
         DISTE=DISTC-NOSERAD+COS(THETA3)*NOSERAD
         DELANG=2.0*THETA3/(CRVGDY1+1)

         DO 20 IY=1,NOSGDY

            X(IX+NOSGDX1,IY,1)=X(IX+NOSGDX1-1,IY,1)+DELDIST

            IF(IY.LE.(INUMN/2))THEN

               Y(IX+NOSGDX1,IY,1)=CUMDELY
               Z(IX+NOSGDX1,IY,1)=DISTD-(CUMDELY/TAN(THETA3))

               CALL STRCH9(DISTE,INUMN/2-1,IY,DELY,C1A,C1B,E1A,E1B,
     *                     NOSGDX2,IX)

               CUMDELY=CUMDELY+DELY
```

130

```
               ELSEIF(IY.GE.(INUMN/2)+1.AND.IY.LE.(INUMN/2)+CRVGDY1+1)THEN

                  ICOUNT=ICOUNT+1
                  DUMMY=THETA3-ICOUNT*DELANG
                  Y(IX+NOSGDX1,IY,1)=(DISTC-NOSERAD)+COS(DUMMY)*NOSERAD
                  Z(IX+NOSGDX1,IY,1)=SIN(DUMMY)*NOSERAD

               ELSE

                  Y(IX+NOSGDX1,IY,1)=Y(IX+NOSGDX1,NOSGDY+1-IY,1)
                  Z(IX+NOSGDX1,IY,1)=-Z(IX+NOSGDX1,NOSGDY+1-IY,1)

               ENDIF

 20      CONTINUE

         PRINT *,'YES3'

C
C    BELOW ARE LISTED SOME MORE VARIABLES USED IN THE PROGRAM:
C
C    BODGDX1 - THE NUMBER OF SECOND DELTA WING GRIDS UP TO THE RECTANGULAR
C              SECTION AND NOT INCLUDING THE FIRST GRID FROM THE NOSE
C    NOSGDX  - THE TOTAL NUMBER OF GRIDS IN THE X DIRECTION OF THE FIRST
C              DELTA WING
C    THETA4  - THE ANGLE FORMED BY THE SECOND DELTA WING
C    DISTZU1 - THE DISTANCE IN THE POSITIVE Z DIRECTION FOR THE FIRST NACA
C              CROSS-SECTION ENCOUNTERED (DISTZU)
C    DISTZD1 - THE SAME DISTANCE ON THE LOWER SURFACE (DISTZD)
C    LEN4    - THE LENGTH IN THE X DIRECTION FROM THE SECOND DELTA WING TO
C              THE REAR RECTANGULAR SECTION
C    LEN5    - THE TOTAL LENGTH OF THE SECOND DELTA WING
C    THETA5  - THE ANGLE ON THE UPPER SURFACE FROM THE CENTERLINE TO THE
C              FIRST NACA SECTION ENCOUNTERED
C    THETA6  - THE SAME ANGLE ON THE LOWER SURFACE
C    DISTA   - THE DISTANCE IN THE Y DIRECTION WITH A NACA CROSS-SECTION
C    CHRD    - CHORD LENGTH OF THE NACA SECTION
C    XDIST   - THE POSITION IN THE X DIRECTION ON THE NACA AIRFOIL
C

C
C    THE SECTION BELOW GENERATES THE GRID FOR THE SECOND DELTA WING
C

         NOSGDX=NOSGDX1+NOSGDX2
         CUMDIST=0.0

         DO 30 IX=NOSGDX+1,NOSGDX+BODGDX1

            IXI=IX-NOSGDX1-NOSGDX2

            CALL STRCH4(LEN4,BODGDX1,IXI,DELDIST,DISTX2)

            CUMDIST=CUMDIST+DELDIST
            X(IX,1,1)=X(IX-1,1,1)+DELDIST
            Y(IX,1,1)=Y(IX-1,1,1)
            Z(IX,1,1)=Z(IX-1,1,1)
            DISTE=LEN2+(CUMDIST/TAN(THETA4))-NOSERAD/SIN(THETA4)
            ICOUNT=0
```

131

```
JCOUNT=0
KCOUNT=0
CUMDELY=0.0

DO 30 IY=2,NOSGDY

    CALL STRCH8(DISTE,INUM/2-1,IY-1,DELY,C2A,C2B,E2A,E2B,
*                BODGDX1,IX-NOSGDX)

    CUMDELY=CUMDELY+DELY
    X(IX,IY,1)=X(IX-1,IY,1)+DELDIST

    IF(IY.LE.(INUM/2))THEN

        Y(IX,IY,1)=Y(IX,IY-1,1)+DELY

    ELSEIF(IY.GE.(INUM/2)+1.AND.IY.LE.(INUM/2)+
*          (CRVGDY-1)/2+1)THEN

        THETA9=ATAN((Y(IX,INUM/2,1)-Y(IX,INUM/2-1,1))/
*              (Z(IX,INUM/2-1,1)-Z(IX,INUM/2,1)))
        DELANG=2.0*THETA9/(CRVGDY+1)
        JCOUNT=JCOUNT+1
        YRAD=Z(IX,INUM/2,1)/SIN(THETA9)
        Y(IX,IY,1)=Y(IX,INUM/2,1)+(COS(THETA9-DELANG*JCOUNT)
*                  *YRAD)-YRAD*COS(THETA9)

    ELSE

        Y(IX,IY,1)=Y(IX,NOSGDY+1-IY,1)

    ENDIF

    CALL NACA006(DISTZU1,DISTZD1,LEN5,CUMDIST)

    THETA5=ATAN((Z(IX,1,1)-DISTZU1)/LEN2)
    THETA6=ATAN((Z(IX,1,1)+DISTZD1)/LEN2)

    IF(IY.LT.((NOSGDY-1)/2)+1)THEN

        IF(Y(IX,IY,1).LE.LEN2)THEN

            Z(IX,IY,1)=Z(IX,1,1)-TAN(THETA5)*Y(IX,IY,1)

        ELSEIF(Y(IX,IY,1).GT.LEN2.AND.IY.LE.(INUM/2))THEN

            DISTA=Y(IX,IY,1)-LEN2
            CHRD=LEN5-TAN(THETA4)*DISTA
            XDIST=CUMDIST-TAN(THETA4)*DISTA

            CALL NACA006(DISTZU,DISTZD,CHRD,XDIST)

            Z(IX,IY,1)=DISTZU

        ELSE

            KCOUNT=KCOUNT+1
            Z(IX,IY,1)=SIN(THETA9-DELANG*KCOUNT)*YRAD

        ENDIF
```

```
            ELSEIF(IY.EQ.((NOSGDY-1)/2)+1)THEN

                KCOUNT=KCOUNT+1
                Z(IX,IY,1)=0.0

            ELSE

                IF(IY.LE.INUM/2+CRVGDY+1)THEN

                    KCOUNT=KCOUNT+1
                    Z(IX,IY,1)=SIN(THETA9-DELANG*KCOUNT)*YRAD

                ELSEIF(Y(IX,IY,1).GT.LEN2)THEN

                    DISTA=Y(IX,IY,1)-LEN2
                    CHRD=LEN5-TAN(THETA4)*DISTA
                    XDIST=CUMDIST-TAN(THETA4)*DISTA

                    CALL NACA006(DISTZU,DISTZD,CHRD,XDIST)

                    Z(IX,IY,1)=DISTZD

                ELSE

                    ICOUNT=ICOUNT+1
                    Z(IX,IY,1)=Z(IX-1,NOSGDY,1)+TAN(THETA6)*Y(IX,IY,1)

                ENDIF

            ENDIF

 30     CONTINUE

        PRINT *,'YES4'

C
C   THIS SECTION COMPUTES THE LAST RECTANGULAR SECTION OF THE WING
C

        CUMDIST=0.0
        INUM2=NOSGDY-SIDGRD
        NOSERAD1=NOSERAD*2.0

        DO 40 IX=NOSGDX+BODGDX1+1,NOSGDX+BODGDX1+BODGDX2

            IF(IX.GE.NOSGDX+BODGDX1+10) INUM2=INUM2+3
            XI=((LEN5-LEN4-NOSERAD)**2.0+
     *          (Z(NOSGDX+BODGDX1,IY,1)-NOSERAD)**2.0)**0.5
            ANG1=ATAN((Z(NOSGDX+BODGDX1,IY,1)-NOSERAD)/
     *          (LEN5-LEN4-NOSERAD))
            ANG2=ACOS(NOSERAD/X1)
            THETA7=PI-ANG1-ANG2
            DELANG=THETA7/2.0
            DISTD=(LEN5-LEN4-NOSERAD+COS(THETA7)*NOSERAD)
            CUMDELY=0.0
            ICOUNT=0
            JCOUNT=0
            KCOUNT=0
            LCOUNT=0
```

```fortran
      MCOUNT=0
      IXI=IX-NOSGDX1-NOSGDX2-BODGDX1

      CALL STRCH4(DISTD,BODGDX2,IXI,DELDIST,DISTX3)

      CUMDIST=CUMDIST+DELDIST

      DO 40 IY=1,NOSGDY

          XI=((LEN5-LEN4-NOSERAD)**2.0+
     *        (Z(NOSGDX+BODGDX1,IY,1)-NOSERAD)**2.0)**0.5
          ANG1=ATAN((Z(NOSGDX+BODGDX1,IY,1)-NOSERAD)/
     *        (LEN5-LEN4-NOSERAD))
          ANG2=ACOS(NOSERAD/XI)
          THETA7=PI-ANG1-ANG2
          DELANG=THETA7/2.0
          DISTD=(LEN5-LEN4-NOSERAD+COS(THETA7)*NOSERAD)
          X(IX,IY,1)=X(IX-1,IY,1)+DELDIST
          DISTE=LEN6-NOSERAD

          IF(IY.LE.((NOSGDY-1)/2)+1)THEN

              IF(IY.LE.(INUM2/2))THEN

                  Y(IX,IY,1)=CUMDELY

                  CALL STRCH8(DISTE,INUM2/2-1,IY,DELY,C3A,C3B,E3A,E3B,
     *                        BODGDX2,IX-NOSGDX-BODGDX1)

                  CUMDELY=CUMDELY+DELY

              ELSEIF(IY.GT.INUM2/2.AND.
     *               IY.LE.INUM2/2+2)THEN

                  ICOUNT=ICOUNT+1
                  DELANG1=(PI/4.0)
                  Y(IX,IY,1)=Y(IX,INUM2/2,1)+
     *                       COS(PI/2.0-ICOUNT*DELANG1)*NOSERAD

              ELSE

                  Y(IX,IY,1)=Y(IX,IY-1,1)

              ENDIF

          ELSE

              JCOUNT=JCOUNT+1
              Y(IX,IY,1)=Y(IX,IY-2*JCOUNT,1)

          ENDIF

          IF(Y(IX,IY,1).LE.LEN2.AND.IY.LE.INUM2/2)THEN

              Z(IX,IY,1)=Z(IX-1,IY,1)-DELDIST/TAN(THETA7)

              IF(Z(IX,IY,1).LT.NOSERAD1)THEN

                  Z(IX,IY,1)=NOSERAD1
```

134

```fortran
            ENDIF

        ELSEIF(Y(IX,IY,1).GT.LEN2.AND.IY.LE.INUM2/2)THEN

            DISTA=Y(IX,IY,1)-LEN2
            CHRD=LEN5-TAN(THETA4)*DISTA
            XDIST=CHRD-(LEN5-LEN4)+CUMDIST

            CALL NACA006(DISTZU,DISTZD,CHRD,XDIST)

            Z(IX,IY,1)=DISTZU
            IF(Z(IX,IY,1).LT.NOSERAD1)THEN

                Z(IX,IY,1)=NOSERAD1

            ENDIF

        ELSEIF(IY.GT.INUM2/2.AND.IY.LE.INUM2/2+2)THEN

            LCOUNT=LCOUNT+1
            Z(IX,IY,1)=Z(IX,INUM2/2,1)-NOSERAD+
     *                 SIN(PI/2-LCOUNT*DELANG1)*NOSERAD

        ELSEIF(IY.GT.INUM2/2+2.AND.IY.LT.(NOSGDY-1)/2+1)THEN

            DIFF=((NOSGDY-1)/2+1)-(INUM2/2+2)
            DEL=Z(IX,INUM2/2+2,1)/DIFF
            Z(IX,IY,1)=Z(IX,IY-1,1)-DEL

        ELSEIF(IY.EQ.((NOSGDY-1)/2)+1)THEN

            Z(IX,IY,1)=0.0

        ELSE

            MCOUNT=MCOUNT+1
            Z(IX,IY,1)=-1.0*Z(IX,IY-2*MCOUNT,1)

        ENDIF

40      CONTINUE

        PRINT *,'YES5'

C
C  THIS SECTION DOES THE REPEATING OF THE TRAILING EDGE GRID TO
C  FORM THE SECTION OF THE SURFACE GRID THAT FORMS THE WAKE
C
        TOT=NOSGDX+BODGDX1+BODGDX2

        DO 50 IX=1,WAKGRD

            CALL STRCH5(WAKLEN,WAKGRD,IX,DELX,DISTX4)

            DO 50 IY=1,NOSGDY

                ITOT=TOT+IX
                X(ITOT,IY,1)=X(ITOT-1,IY,1)+DELX
                Y(ITOT,IY,1)=Y(ITOT-ix,IY,1)
```

```fortran
              Z(ITOT,IY,1)=Z(ITOT-ix,IY,1)

 50      CONTINUE

         PRINT *,'YES6'

C
C   PRINT THE DATA TO A FILE
C
C        DO 11 I=40,42
C        DO 11 I=1,NOSGDX+BODGDX1+BODGDX2+WAKEGRD
C           DO 11 J=1,NOSGDY
C              PRINT *,I,J,X(I,J,1),Y(I,J,1),Z(I,J,1)
C 11     CONTINUE

         IZ=1
         IX=1
         IY=163
         REWIND 3
         WRITE(3) IY-IX+1,NOSGDY,IZ
         WRITE(3) ((X(I,J,1),I=IX,IY),J=1,NOSGDY),
     *            ((Y(I,J,1),I=IX,IY),J=1,NOSGDY),
     *            ((Z(I,J,1),I=IX,IY),J=1,NOSGDY)

         CLOSE(UNIT=10)
         CLOSE(UNIT=11)

         STOP

         END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                        C
C  THIS IS THE SUBROUTINE NACA006 AND IT EVALUTATES THE Z DIRECTION VALUES  C
C            THAT ARE ASSIGNED TO THAT SPECIFIC CROSS-SECTION            C
C                                                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


         SUBROUTINE NACA006(ZPOS,ZNEG,CHORD,XVAL)

         DIMENSION XPC(26),YPC(26)

         DATA XPC/0.0,0.5,0.75,1.25,2.5,5.0,7.5,10.0,15.0,20.0,25.0,30.0,
     *          35.0,40.0,45.0,50.0,55.0,60.0,65.0,70.0,75.0,80.0,85.0,
     *          90.0,95.0,100.0/
         DATA YPC/0.0,0.494,0.596,0.754,1.024,1.405,1.692,1.928,2.298,
     *          2.572,2.772,2.907,2.981,2.995,2.919,2.775,2.575,2.331,
     *          2.050,1.740,1.412,1.072,0.737,0.423,0.157,0.0/

         X=XVAL/CHORD*100.0

         DO 10 I=1,25

            IF(X.GT.XPC(I).AND.X.LT.XPC(I+1))THEN

               IVAL=I
```

```fortran
            GO TO 99

        ELSE

            CONTINUE

        ENDIF

10    CONTINUE

99    VAL1=X-XPC(IVAL)
      VAL2=XPC(IVAL+1)-XPC(IVAL)
      VAL3=YPC(IVAL+1)-YPC(IVAL)
      ZVAL=(VAL1*VAL3/VAL2)+YPC(IVAL)
      ZPOS=ZVAL*CHORD/100.0
      ZNEG=-1.0*ZPOS

      RETURN

      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                              C
C  THIS SUBROUTINE STRETCHES THE GRID IN THE X DIRECTION       C
C                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```fortran
      SUBROUTINE STRCH1(DIA,IDIV,IA,DELDIST)

      YDIST=DIA/IDIV

      IF(IA.LE.IDIV/2)THEN

          IB=IA
          IC=1

      ELSE

          IB=IDIV-IA
          IC=-1

      ENDIF

      IF(IA.GE.IDIV/2.AND.IA.LE.IDIV/2+1)THEN

          DELDIST=((DIA/2.0)**2.0-((IDIV/2-1)*YDIST)**2.0)**0.5

      ELSE

          DIST1=((DIA/2.0)**2.0-(IB*YDIST)**2.0)**0.5
          DIST2=((DIA/2.0)**2.0-((IB-IC)*YDIST)**2.0)**0.5
          DELDIST=ABS(DIST1-DIST2)

      ENDIF

      RETURN
```

137

```
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                         C
C  THIS IS THE SUBROUTINE FOR STRETCHING IN THE Y DIRECTION  C
C                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE STRCH2(RAD,JDIV,IA,DELY)

      YDIST=RAD/JDIV

      IF(IA.EQ.1)THEN

          DELY=((RAD**2.0)-((JDIV-IA)*YDIST)**2.0)**0.5

      ELSE

          DIST1=((RAD**2.0)-((JDIV-IA)*YDIST)**2.0)**0.5
          DIST2=((RAD**2.0)-((JDIV-IA+1)*YDIST)**2.0)**0.5
          DELY=ABS(DIST1-DIST2)

      ENDIF

      RETURN

      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                         C
C  THIS SUBROUTINE IS FOR LINEAR STRETCHING IN THE X DIRECTION C
C                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE STRCH4(DIA,IDIV,IA,DELDIST,XVAR)

      TOT=0.0
      SUBTOT=1.0

      DO 10 I=1,IDIV/2-1

          SUBTOT=XVAR*SUBTOT
          TOT=TOT+SUBTOT

10    CONTINUE

      SUBTOT=(DIA/2)/(TOT+1)

      IF(IA.LE.IDIV/2)THEN

          IB=IA

      ELSE
```

```
            IB=IA-(IDIV/2)

        ENDIF

        IF(IB.EQ.1)THEN

            DELDIST=SUBTOT

        ELSE

            DO 20 I=1,IB-1

                SUBTOT=XVAR*SUBTOT

20      CONTINUE

            DELDIST=SUBTOT

        ENDIF

        RETURN

        END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                               C
C  THIS SUBROUTINE IS FOR LINEAR STRETCHING IN THE WAKE X DIR   C
C                                                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


        SUBROUTINE STRCH5(DIA,IDIV,IA,DELDIST,XVAR)

        TOT=0.0
        SUBTOT=1.0

        DO 10 I=1,IDIV-1

            SUBTOT=XVAR*SUBTOT
            TOT=TOT+SUBTOT

10      CONTINUE

        SUBTOT=DIA/(TOT+1)

        IF(IA.EQ.1)THEN

            DELDIST=SUBTOT

        ELSE

            DO 20 I=1,IA-1

                SUBTOT=XVAR*SUBTOT

20      CONTINUE
```

```fortran
          DELDIST=SUBTOT

      ENDIF

      RETURN

      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                             C
C  THIS SUBROUTINE IS FOR LINEAR STRETCHING IN THE Y DIRECTION C
C                                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


      SUBROUTINE STRCH6(DIA,IDIV,IA,DELDIST,XVAR1,XVAR2,INUM,IY)

      TOT=0.0
      SUBTOT=1.0
      XVAR=((IY-1.0)/(INUM-1.0))*(XVAR2-XVAR1)+XVAR1

      DO 10 I=1,IDIV-1

         SUBTOT=XVAR*SUBTOT
         TOT=TOT+SUBTOT

10    CONTINUE

      SUBTOT=DIA/(TOT+1)

      IB=IDIV-IA-1

      IF(IB.EQ.1)THEN

         DELDIST=SUBTOT

      ELSE

         DO 20 I=1,IB-1

            SUBTOT=XVAR*SUBTOT

20       CONTINUE

         DELDIST=SUBTOT

      ENDIF

      RETURN

      END

******************************************************************
******************************************************************
*  THIS IS A SUBROUTINE FOR EXPONENTIAL STRETCHING IN THE Y-DIRECTION  *
```

```
******************************************************************
******************************************************************


      SUBROUTINE STRCH7(DIA,IDIV,IA,DELDIST,CON1,CON2,EX1,EX2,KX,KA)
      CON=(KA-1.0)/(KX-1.0)*(CON2-CON1)+CON1
      EX=(KA-1.0)/(KX-1.0)*(EX2-EX1)+EX1
      DIA1=CON*DIA**EX
      DELY=DIA1/IDIV


      DISTY1=IA*DELY
      DISTY2=(IA-1.0)*DELY
      VAL1=(DISTY1/CON)**(1.0/EX)
      VAL2=(DISTY2/CON)**(1.0/EX)
      DELDIST=VAL1-VAL2

      RETURN
      END

******************************************************************
******************************************************************
*   THIS IS A SUBROUTINE FOR SINEUSOIDALLY STRETCHING THE Y-DIRECTION *
******************************************************************
******************************************************************


      SUBROUTINE STRCH8(DIA,IDIV,IA,DELDIST,C1,C2,E1,E2,KX,KA)

      CON=(KA-1.0)/(KX-1.0)*(C2-C1)+C1
      EX=(KA-1.0)/(KX-1.0)*(E2-E1)+E1
      RAD45=.785398163

      DELY=RAD45/IDIV
      DISTY1=IA*DELY+RAD45
      DISTY2=(IA-1.0)*DELY+RAD45
      VAL1=CON*SIN(DISTY1)**EX
      VAL2=CON*SIN(DISTY2)**EX
      DIST=VAL1-VAL2
      DELDIST=DIST*DIA/(1.0-SIN(RAD45))

      RETURN

      END



******************************************************************
******************************************************************
*   THIS IS A SUBROUTINE FOR SINEUSOIDALLY STRETCHING THE Y-DIRECTION *
******************************************************************
******************************************************************


      SUBROUTINE STRCH9(DIA,IDIV,IA,DELDIST,C1,C2,E1,E2,KX,KA)

      CON=(KA-1.0)/(KX-1.0)*(C2-C1)+C1
      EX=(KA-1.0)/(KX-1.0)*(E2-E1)+E1
      PI=4.0*ATAN(1.0)
```

```
RAD45=PI/4.0
RAD60=PI/3.0
RAD90=PI/2.0

RADSTP=(RAD90-RAD60)/(KX-1)

DELY=(((RADSTP*(KA-1))+RAD60)-RAD45)/(IDIV)
DISTY1=(IA-1)*DELY+RAD45
DISTY2=(IA)*DELY+RAD45
VAL1=CON*SIN(DISTY1)**EX
VAL2=CON*SIN(DISTY2)**EX
DIST=VAL2-VAL1
DELDIST=DIST*DIA/(SIN(RAD45+DELY*IDIV)-SIN(RAD45))

RETURN

END
```

APPENDIX F.   ADDITIONAL SOURCE CODE

```
C
C THIS PROGRAM REMOVES THE FIRST THREE POINTS OF THE HEMISPHERE, BRINGS
C THE APEX TO A POINT, RENUMBERS THE GRID POINTS AND FINALLY DOUBLE THE
C THICKNESS OF THE SURFACE GRID IN THE Z-DIRECTION
C
      DIMENSION X(140,240,60),Y(140,240,60),Z(140,240,60),
     *          XX(140,240,60),YY(140,240,60),ZZ(140,240,60)

      READ(3) II,JJ,KK
      READ(3) (((X(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Y(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Z(J,K,L),J=1,II),K=1,JJ),L=1,KK)


      DO 10 I=1,II-3

        DO 10 J=1,JJ

          IF(I.EQ.1)THEN

            XX(1,J,1)=0.0
            YY(1,J,1)=0.0
            ZZ(1,J,1)=0.0

          ELSE
            XX(I,J,1)=X(I+3,J,1)
            YY(I,J,1)=Y(I+3,J,1)
            ZZ(I,J,1)=Z(I+3,J,1)*2.0

          ENDIF

10    CONTINUE

      REWIND 3

      WRITE(3) II-3,JJ,1
      WRITE(3) ((XX(J,K,1),J=1,II-3),K=1,JJ),
     *         ((YY(J,K,1),J=1,II-3),K=1,JJ),
     *         ((ZZ(J,K,1),J=1,II-3),K=1,JJ)


      STOP

      END
```

```fortran
C
C THIS PROGRAM AGAIN DELETES THREE GRID POINTS FROM THE NOSE OF THE
C SURFACE GRID.  THIS PARTICULAR PROGRAM THEN CLOSES THE APEX OF THE
C SURFACE GRID TO A POINT AND FINALLY RENUMBERS THE GRID FOR FIELD GRID
C GENERATION
C

      DIMENSION X(150,240,60),Y(150,240,60),Z(150,240,60),
     *          XX(150,240,60),YY(150,240,60),ZZ(150,240,60)

      READ(3) II,JJ,KK
      READ(3) (((X(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Y(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Z(J,K,L),J=1,II),K=1,JJ),L=1,KK)


      DO 10 I=1,II-3

         DO 10 J=1,JJ

            IF(I.EQ.1)THEN

               XX(1,J,1)=0.0
               YY(1,J,1)=0.0
               ZZ(1,J,1)=0.0

            ELSE

               XX(I,J,1)=X(I+3,J,1)
               YY(I,J,1)=Y(I+3,J,1)
               ZZ(I,J,1)=Z(I+3,J,1)

            ENDIF

10    CONTINUE

      REWIND 3

      WRITE(3) II-3,JJ,1
      WRITE(3) ((XX(J,K,1),J=1,II-3),K=1,JJ),
     *         ((YY(J,K,1),J=1,II-3),K=1,JJ),
     *         ((ZZ(J,K,1),J=1,II-3),K=1,JJ)

      STOP

      END
```

```
C
C THIS PROGRAM READS THE FIELD GRID AND JUST CHANGES THE Z-DIRECTION
C VALUE BY DOUBLING THE THICKNESS
C
      DIMENSION X(150,240,60),Y(150,240,60),Z(150,240,60),
     *          XX(150,240,60),YY(150,240,60),ZZ(150,240,60)

      READ(3) II,JJ,KK
      READ(3) (((X(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Y(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Z(J,K,L),J=1,II),K=1,JJ),L=1,KK)


      DO 10 I=1,II

         DO 10 J=1,JJ

            IF(I.EQ.1)THEN

               XX(1,J,1)=X(I,J,1)
               YY(1,J,1)=0.0
               ZZ(1,J,1)=0.0

            ELSE

               XX(I,J,1)=X(I,J,1)
               YY(I,J,1)=Y(I,J,1)
               ZZ(I,J,1)=Z(I,J,1)*2.0

            ENDIF

10    CONTINUE

      REWIND 3

      WRITE(3) II,JJ,1
      WRITE(3) ((XX(J,K,1),J=1,II),K=1,JJ),
     *         ((YY(J,K,1),J=1,II),K=1,JJ),
     *         ((ZZ(J,K,1),J=1,II),K=1,JJ)


      STOP

      END
```

```fortran
C
C THIS PROGRAM ALSO DELETES THE FIRST THREE POINTS OF THE HEMISPHERE ON
C THE SURFACE GRID.  BUT THIS PROGRAM ALSO DOUBLES THE THICKNESS OF THE
C SURFACE GRID IN THE Z-DIRECTION ONLY
C
      DIMENSION X(150,240,60),Y(150,240,60),Z(150,240,60),
     *          XX(150,240,60),YY(150,240,60),ZZ(150,240,60)

      READ(3) II,JJ,KK
      READ(3) (((X(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Y(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Z(J,K,L),J=1,II),K=1,JJ),L=1,KK)


      DO 10 I=1,II-10

         DO 10 J=1,JJ

            IF(I.EQ.1)THEN

               XX(1,J,1)=X(I+10,J,1)
               YY(1,J,1)=0.0
               ZZ(1,J,1)=0.0

            ELSE

               XX(I,J,1)=X(I+10,J,1)
               YY(I,J,1)=Y(I+10,J,1)
               ZZ(I,J,1)=Z(I+10,J,1)*2.0

            ENDIF

10    CONTINUE

      REWIND 3

      WRITE(3) II-10,JJ,1
      WRITE(3) ((XX(J,K,1),J=1,II-10),K=1,JJ),
     *         ((YY(J,K,1),J=1,II-10),K=1,JJ),
     *         ((ZZ(J,K,1),J=1,II-10),K=1,JJ)

      STOP

      END
```

```
C
C  THIS PROGRAM READS THE FINISHED SURFACE GRID AND DELETES THE FIRST
C  THREE POINTS THAT WERE CONSTRUCTED BY THE HEMISPHERE AND THEN
C  RENUMBERS THE GRID FOR USE BY THE FIELD GRID GENERATOR
C

      DIMENSION X(120,240,60),Y(120,240,60),Z(120,240,60),
     *          XX(120,240,60),YY(120,240,60),ZZ(120,240,60)

      READ(3) II,JJ,KK
      READ(3) (((X(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Y(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *        (((Z(J,K,L),J=1,II),K=1,JJ),L=1,KK)


      DO 10 I=1,II-10

         DO 10 J=1,JJ

            IF(I.EQ.1)THEN

               XX(1,J,1)=X(I+10,J,1)
               YY(1,J,1)=0.0
               ZZ(1,J,1)=0.0

            ELSE

               XX(I,J,1)=X(I+10,J,1)
               YY(I,J,1)=Y(I+10,J,1)
               ZZ(I,J,1)=Z(I+10,J,1)

            ENDIF

10    CONTINUE

      REWIND 3

      WRITE(3) II-10,JJ,1
      WRITE(3) ((XX(J,K,1),J=1,II-10),K=1,JJ),
     *         ((YY(J,K,1),J=1,II-10),K=1,JJ),
     *         ((ZZ(J,K,1),J=1,II-10),K=1,JJ)


      STOP

      END
```

```
C
C THIS IS A PROGRAM THAT MANUALLY ADJUSTS THE FIELD GRID NOSE REGION
C FOR THE CYLINDRICAL GRID.  IT REDISTRIBUTES THE X VALUE ALONG THE
C SINGULARITY OF THE NOSE.  IT ALSO RESIGNS THE Y AND Z VALUES FOR THE
C FIELD GRID.
C
      DIMENSION X(117,240,64),Y(117,240,64),Z(117,240,64),
     *          XX(14,240,64),YY(14,240,64),ZZ(14,240,64),
     *          XXX(130,240,64),YYY(130,240,64),ZZZ(130,240,64)

      OPEN(UNIT=10,FILE='ddwnos.in',STATUS='OLD')

      READ(10,*) DIA,XVAR
      READ(24) II,JJ,KK
      READ(24)  (((X(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *          (((Y(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *          (((Z(J,K,L),J=1,II),K=1,JJ),L=1,KK)

      DO 10 I=1,JJ

         DO 10 J=1,KK

            XX(1,I,J)=X(1,I,J)
            YY(1,I,J)=Y(1,I,J)
            ZZ(1,I,J)=Z(1,I,J)

10    CONTINUE

      DO 13 I=2,14

         CALL STRCH(DIA,13,I-1,DELDIST,YVAR)

         DO 13 J=1,JJ

            DO 13 K=1,KK

               IF(I.EQ.2)THEN

                  XX(I,J,K)=DELDIST

               ELSE

                  XX(I,J,K)=XX(I-1,J,K)+DELDIST

               ENDIF
13    CONTINUE

      DO 11 I=2,14

         DO 11 J=1,JJ

            DO 11 K=1,KK

               IF(K.EQ.1)THEN

                  YY(I,J,K)=0.0
                  ZZ(I,J,K)=0.0
```

```
              ELSE

                  YY(I,J,K)=Y(1,J,K)-Y(1,J,1)
                  ZZ(I,J,K)=Z(1,J,K)-Z(1,J,1)

              ENDIF

11    CONTINUE

      DO 17 I=1,130

         DO 17 J=1,JJ

            DO 17 K=1,KK

               IF(I.LE.13)THEN

                  XXX(I,J,K)=XX(15-I,J,K)
                  YYY(I,J,K)=YY(I,J,K)
                  ZZZ(I,J,K)=ZZ(I,J,K)

               ELSE

                  XXX(I,J,K)=X(I-13,J,K)
                  YYY(I,J,K)=Y(I-13,J,K)
                  ZZZ(I,J,K)=Z(I-13,J,K)

               ENDIF
17    CONTINUE

      REWIND 34

      WRITE(34) 130,JJ,KK
      WRITE(34) (((XXX(J,K,L),J=1,130),K=1,JJ),L=1,KK),
     *          (((YYY(J,K,L),J=1,130),K=1,JJ),L=1,KK),
     *          (((ZZZ(J,K,L),J=1,130),K=1,JJ),L=1,KK)


      STOP

      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  THIS IS THE SUBROUTINE THAT REDISTRIBUTES THE GRID POINTS USING A  C
C  LINEAR TYPE DISTRIBUTION.                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


      SUBROUTINE STRCH(DIA,IDIV,IA,DELDIST,XVAR)

      TOT=0.0
      SUBTOT=1.0

      DO 10 I=1,IDIV-1

         SUBTOT=XVAR*SUBTOT
         TOT=TOT+SUBTOT
```

```
10    CONTINUE

      SUBTOT=DIA/(TOT+1)

      IF(IA.EQ.1)THEN

        DELDIST=0.0-SUBTOT

      ELSE

        DO 20 I=1,IA-1

          SUBTOT=XVAR*SUBTOT

20      CONTINUE

        DELDIST=0.0-SUBTOT

      ENDIF

      RET"RN

      END
```

```
C
C THIS IS A PROGRAM THE READS THE FINISHED SURFACE GRID AND CAN
C BUILD A FILE OF ANY PARTICULAR YZ-PLANE CROSS SECTION THAT IS DESIRED.
C THE DESIRED PLANE IS CHOSEN BY CHANGING THE "I" VARIABLE.
C

      DIMENSION X(117,240,60),Y(117,240,60),Z(117,240,60),
     *          XX(1,240,60),YY(1,240,60),ZZ(1,240,60)

      READ(24) II,JJ,KK
      READ(24) (((X(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *         (((Y(J,K,L),J=1,II),K=1,JJ),L=1,KK),
     *         (((Z(J,K,L),J=1,II),K=1,JJ),L=1,KK)

      I=89

      DO 10 J=1,JJ

         DO 10 K=1,KK

            XX(1,J,K)=X(I,J,K)
            YY(1,J,K)=Y(I,J,K)
            ZZ(1,J,K)=Z(I,J,K)

10    CONTINUE

      REWIND 37

      WRITE(37) 1,JJ,KK
      WRITE(37) ((XX(1,K,L),K=1,JJ),L=1,KK),
     *          ((YY(1,K,L),K=1,JJ),L=1,KK),
     *          ((ZZ(1,K,L),K=1,JJ),L=1,KK)

      STOP

      END
```

LIST OF REFERENCES

1. Cumming. R. M., Rizk, Y. M., Schiff, L. B. and Chaderjian, N. M., *Navier-Stokes Prediction of the Flowfield Around the F-18 (HARV) Wing and Fuselage at Large Incidence*, AIAA Paper 90-0099 presented at Aerospace Science Meeting in Reno, NV, January, 1990.

2. Hoeijmakers, H. W. M., *Computational Vortex Flow in Aerodynamics*, AGARD-CCP 342, 1983.

3. Schlichting. H., *Boundary Layer Theory*, McGraw-Hill Book Co., Inc., New York, NY, 1987.

4. Thompson, Joe F., Warsi, A. U. A., and Mastin, Wayne C., *Numerical Grid Generation Foundations and Applications*, Elsevier Science Publishing Co., Inc., New York. NY, 1985.

5. Hirsch. C., *Numerical Computation of Internal and External Flows*, John Wiley and Sons Ltd., New York, NY, 1988.

6. Ying. S. X., Steger, J. L., Schiff, L. B., and Bagannoff, D., *Numerical Simulation of Unsteady, Viscous High-Angle-of-Attack Flows Using a Partially Flux-Split Algorithm*, AIAA Paper 86-2179, 1986.

7. Peckham. D. H. and Atkinson, S. A., *Preliminary Results of Low Speed Tunnel Tests on a Gothic Wing of Aspect Ratio 1,"* Aeronaut. Res. Counc., CP 508, 1957, Paper 16.

8. Elle, B. J., *On the Breakdown at High Incidence of the Leading Edge Vortices on Delta Wings*, J. R. Aeronaut. Soc., 65, 1960, pp. 491-493.

9. Lambourne, N. C. and Bryer, D. W., *The Bursting of Leading-Edge Vortices - Some Observations and Discussion of the Phenomenon*, Aeronaut. REs. Counc., R&M 3282, 1961, pp.36-48.

10. Harvey, J. K., *Some Observations of the Vortex Breakdown Phenomenon*, J. Fluid Mech., No. 14, 1962, pp.589-592.

11. Pritchard, W. G., *Solitary Waves in Rotating Fluids*, J. Fluid Mech., Vol. 42, 1970, pp. 61-83.

12. Sarpkaya, T., *Vortex Breakdown in Swirling Conical Flows*, AIAA J., No. 9, 1972, pp. 1792-1799.

13. Sarpkaya, T., *The Effect of Adverse Pressure Gradient on Vortex Breakdown*, AIAA J., No. 12, 1974, pp. 602-607.

14. Hummel, D., *On the Vortex Formation over a Slender Wing at Large Angles of Incidence*, AGARD-CP-247, 15, Norway, 4-6 October, 1978.

15. Faler, J. H. and Leibovitch, S., *An Experimental Map of the Internal Structure of a Vortex Breakdown*, J. Fluid Mech., Vol. 86, 1978, pp. 313-335.

16. Payne, F. M. and Nelson, R. C., *An Experimental Investigation of Vortex Breakdown on Delta Wings*, Vortex Flow Aerodynamics, NASA CP-2416, 1985, pp. 135-161.

17. Krause, Aachen E., and Liu, C. H., *Numerical Studies of Incompressible Flow Around Delta and Double- Delta Wings*, Z. Flugwiss. Weltraumforsch. 13, 1989, pp. 291-301.

18. Cunningham, A. M., Jr. and Boer, R. G., *Low-Speed Unsteady Aerodynamics of a Pitching Straked Wing at High Incidence - Part 1: Test Program, Part 2: Harmonic Analysis*, General Dynamics Corp., AIAA, January 13, 1988.

19. Anderson, John D., Jr., *Fundamentals of Aerodynamics*, McGraw-Hill Book Co., Inc., New York, NY, 1984.

154

20. Ekaterinaris, John A., and Schiff, Lewis B., *Vortical Flows over Delta Wings and Numerical Prediction of Vortex Breakdown*, AIAA-90-0102, January, 1990.

21. Baldwin, B. S. and Lomax, H., *Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows*, AIAA Paper 78-257 presented at AIAA 16th Aerospace Sciences Meeting. Huntsville, AL., January 16-18, 1978.

22. Degani, D. and Schiff, L. B., *Computation of Turbulent Supersonic Flows around Pointed Bodies Having Crossflow Separation*, Journal of Computational Physics, Vol. 66, No. 1, Academic Press, New York, NY., September, 1986.

23. Cebeci, T., *Calculation of Compressible Turbulent Boundary Layers with Heat and Mass Transfer*, AIAA Paper 70-140 presented at AIAA 3rd Fluid and Plasma Dynamics Conference, Los Angeles, CA., June 29 - July 1, 1977.

24. Luh, R. C.-C. and Lombard, C. K., *FASTWO - A 2-D Interactive Algebraic Grid Generator*, AIAA Paper 88-0516 presented at AIAA 28th Aerospace Sciences Meeting. Reno, NV., January 11-14, 1988.

25. Eiseman, Peter R., *Grid Generation for Fluid Mechanics Computations*, Ann. Rev. Fluid Mech. 17, 1985, pp. 487-522.

26. Sorenson, Reese L., *The 3DGRAPE Book: Theory, Users' Manual, Examples*, NASA Ames Research Center, Moffett Field, CA, July 1989.

27. Walatka, Pamela P., and Buning, Pieter G., *PLOT3D User's Manual*, NASA Technical Memorandum 101067, 1989.

28. Gato, W. and Masiello, M. F., *Innovative Aerodynamics: The Sensible Way of Restoring Growth Capability to the EA-6B Prowler*, AIAA-87-2362 paper presented at AIAA 5th Applied Aerodynamics Conference, Monterey, CA., August 17-19, 1987.

29. Thomas, J. L., Taylor, S. L. and Anderson, W. K., *Navier-Stokes Computations of Vortical Flows over Low Aspect Ratio Wings*, AIAA Paper 88-0317 presented at 25th Aerospace Science Meeting. Reno, NV, January, 1988.

30. Kjelgaard, O. and Sellers, W. L., *Detailed Flowfield Measurements over a 75 degree Swept Delta Wing for Code Validation,* AGARD Symposium on Validation of CFD, Lisbon, Portugal, May 2-5, 1988.

31. Hawk, J., Barnett, R. and O'Neil P., *Investigation of High Angle of Attack Vortical Flow Over Delta Wings,* AIAA Paper 90-0101 presented at AIAA 28th Aerospace Sciences Meeting, Reno, NV., January 8-11, 1990.

32. Kegelman, J. T. and Roos, F. W., *The Flowfields of Bursting Vortices Over Moderately Swept Delta Wings,* AIAA Paper 90-0599 presented at AIAA 28th Aerospace Sciences Meeting, Reno, NV., January 8-11, 1990.

INITIAL DISTRIBUTION LIST

10. Dr. R. M. Howard   1
    Department of Aeronautics and Astronautics, Code 67HO
    Naval Postgraduate School
    Monterey, CA 93943-5002

11. Dr. S. K. Hebbar   1
    Department of Aeronautics and Astronautics, Code 67HB
    Naval Postgraduate School
    Monterey, CA 93943-5002

12. Dr. R. E. Ball   1
    Department of Aeronautics and Astronautics, Code 67BP
    Naval Postgraduate School
    Monterey, CA 93943-5002

13. Dr. T. Sarpkaya   1
    Department of Mechanical Engineering, Code 69SL
    Naval Postgraduate School
    Monterey, CA 93943-5002

14. Dr. G. Hobsen   1
    Department of Aeronautics and Astronautics, Code 67
    Naval Postgraduate School
    Monterey, CA 93943-5002

15. Dr. J. A. Ekaterinaris   10
    NASA Ames Research Center (M.S. 258-1)
    Moffett Field, CA 94035

16. Dr. M. S. Chandrasekhara   1
    NASA Ames Research Center (M.S. 260-1)
    Moffett Field, CA 94035

17. Dr. L. W. Carr   1
    NASA Ames Research Center (M.S. 260-1)
    Moffett Field, CA 94035

18. Dr. L. B. Schiff   1
    NASA Ames Research Center (M.S. 258-1)
    Moffett Field, CA 94035

19. Mr. D. P. Bencze   1
    Chief, Applied Aerodynamics Branch
    NASA Ames Research Center (M.S. 227-6)
    Moffett Field, CA 94035

20. Dr. S. Davis   1
    NASA Ames Research Center (M.S. 260-1)
    Moffett Field, CA 94035

21. Mr. T. S. Momiyama   1
    Director, Aircraft Division
    Code AIR-931
    Naval Air Systems Command
    Washington, D. C., 20361-9320

22. Mr. B. Neuman   1
    Aircraft Division
    Code AIR-931
    Naval Air Systems Command
    Washington, D. C., 20361-9320

23. Mr. G. Derderian   1
    Aircraft Division
    Code AIR-931
    Naval Air Systems Command
    Washington, D. C., 20361-9320

24. Ms. L. Cowles   1
    Naval Air Development Center, Code 60C
    Street Road
    Warminster, PA 18974-5000

25. Dr. W. Tseng   1
    Naval Air Development Center
    Street Road
    Warminster, PA 18974-5000

26. Mr. D. Findlay   1
    Naval Air Development Center
    Street Road
    Warminster, PA 18974-5000

27. Mr. M. Walters   1
    Naval Air Development Center
    Street Road
    Warminster, PA 18974-5000

28. Mr. W. J. King   1
    Technology Area Manager, Aircraft
    Code 212
    Office of Naval Technology
    Arlington, VA 22217-5000

29. Dr. T. Cebeci   1
    Professor and Chairman
    Dept. of Aerospace Engineering
    California State University
    Long Beach. CA 90840